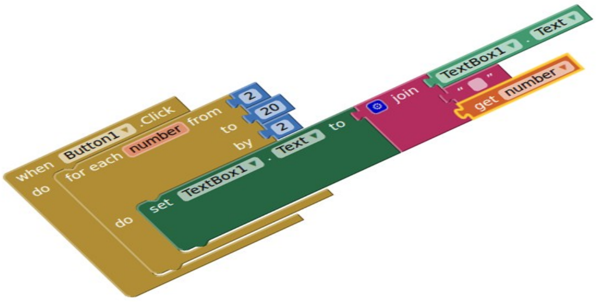
# Introducció bàsica a la creació de programes Apps per dispositius Smartphone i Tablet Android mitjançant l'IDE educatiu AppInventor 2



© Artur Guillamet Sabaté
Novembre 2014

# Introducció i Exercicis bàsics amb AppInventor 2

**INDEX:**

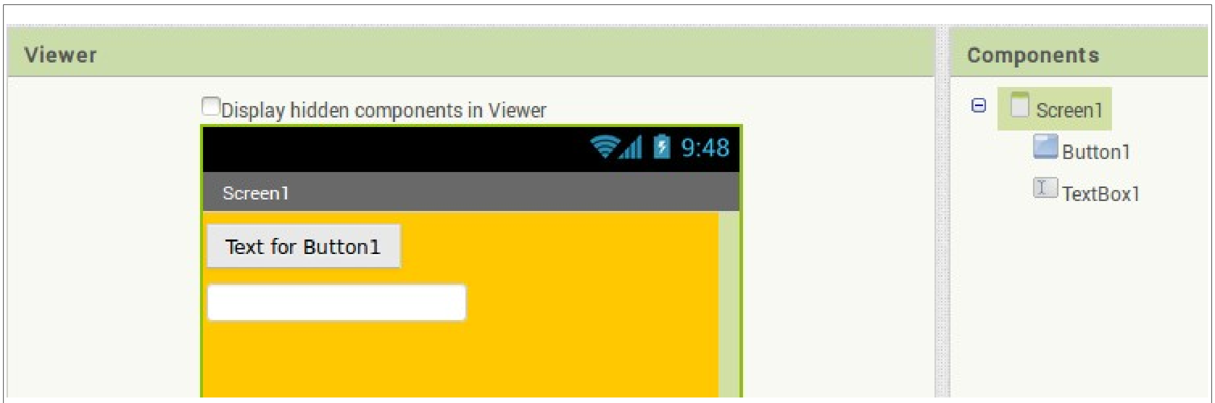

| | Pàg. |
|---|---|
| 1.- Introducció | 2 |
| 2.- Zones de treball | 2 |
| 3.- Interacció amb l'usuari | 3 |
| 4.- Tipus de dades | 3 |
| 5.- Variables | 3 |
| 6.- Operacions aritmètiques | 4 |
| 7.- Operacions lògiques | 4 |
| 8.- Operacions amb cadenes alfanumèriques | 4 |
| 9.- Estructures condicionals | 5 |
| 10.- Estructures iteratives | 5 |
| 11.- Llistes | 6 |
| 12.- Estructures de tractament de llistes | 6 |
| 13.- Pràctica no. 1. Hola món. | 6 |
| 14.- Pràctica no. 2. Compara edat. | 7 |
| 15.- Pràctica no. 3. Llista nombres parells. | 7 |
| 16.- Pràctica no. 4. Daus. | 8 |
| 17.- Pràctica no. 5. Taules de multiplicar. | 10 |
| 18.- Pràctica no. 6. Equacions de $2^n$ grau. | 12 |
| 19.- Pràctica no. 7. Ping-pong. | 16 |
| 20.- Pràctica no. 8. Polígons. | 18 |
| 21.- Exercicis | 24 |

Exercici 1.- Operacions amb frases.
Exercici 2.- Càlculs estadístics.
Exercici 3.- Xocs aleatoris d'sprites.
Exercici 4.- Sensors posició geogràfica i bases de dades.
Exercici 5.- Àudio Player

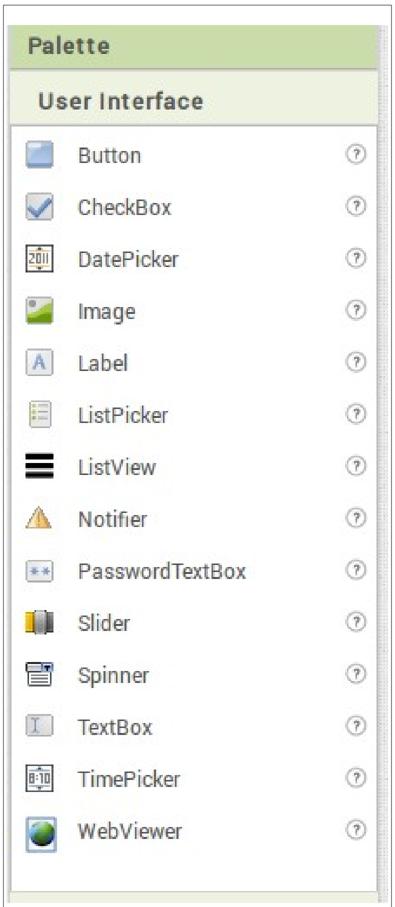
# 1.- Introducció

AppInventor és un entorn de programació visual per blocs, senzill i amb orientació pedagògica, per a dispositius Smartphones i Tablets amb el S.O. Android. Es troba disponible a la web de l'Institut Tecnològic de Massachusetts, MIT: http://ai2.appinventor.mit.edu

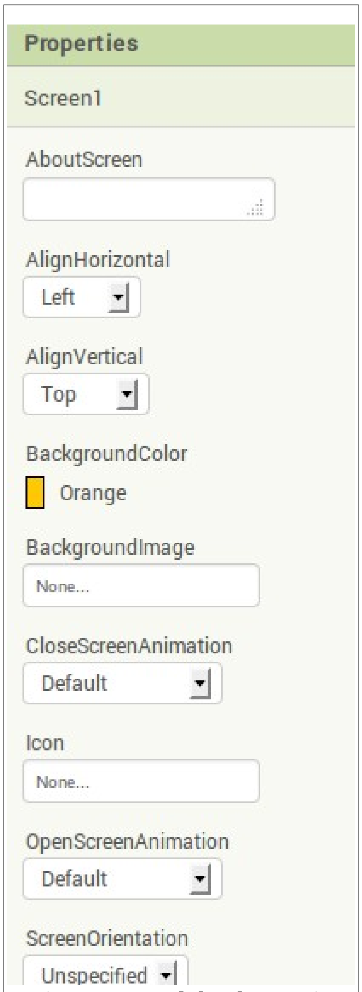
# 2.- Zones de treball



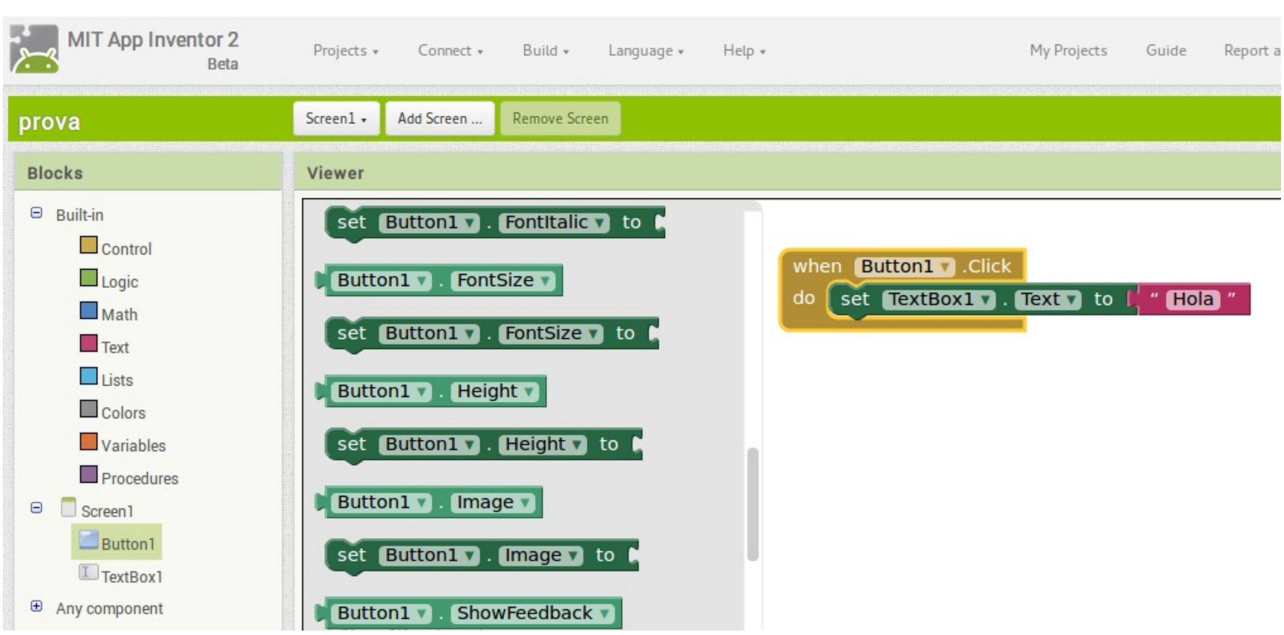

*(Disseny de la interfície)*




*(Paleta d'objectes)*




*(Propietats dels objectes)*




*(Disseny lògic, programa)*

## 3.- Interacció amb l'usuari

Els programes informàtics requereixen de la introducció de dades per part dels usuaris. Aquestes es processaran segons la funcionalitat del programa i es mostrarà els resultats a l'usuari. Si aquestes dades són numèriques i/o textuals, la interacció és farà mitjançant components de la paleta.

**Entrada de dades**: Introduirem les dades mitjançant els TextBox (Quadres de text) .

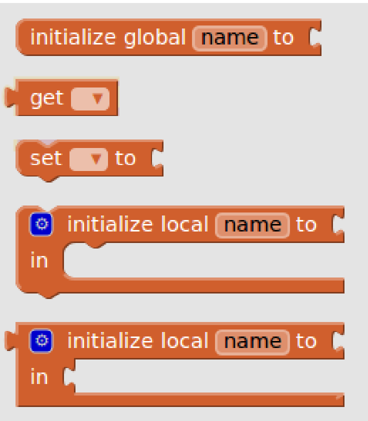
**Sortida de dades**: Obtindrem informació textual mitjançant els TextBox i els Label (Etiqueta).

Iniciarem l'execució mitjançant l'existència d'un event "Click" sobre un Button (Botó)

Un esdeveniment és una acció efectuada sobre un objecte de la interfície del programa (un click, doble click, passar-hi amb el ratolí, etc).
Els blocs de programa iniciaran la seva execució quan s'hagi activat l'esdeveniment adequat sobre l'objecte a qui correspongui.

## 4.- Tipus de dades

De forma genèrica, podem considerar que les dades es poden classificar en dos tipus: Dades numèriques i Dades alfanumèriques. Dins de les dades numèriques hi han altres subtipus.



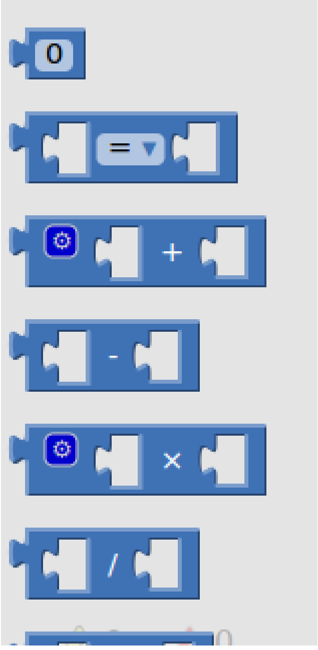

## 5.- Variables

Les variables són elements contenidors de informació. Estan definides mitjançant un nom i es pot assignar-hi (set) o llegir-hi (get) valors.

Poden ser globals o locals. Si són globals, seran accessibles des de qualsevol lloc del programa, en canvi, si són locals, solament seran accessibles des de la funció o rutina on s'estiguin utilitzant.

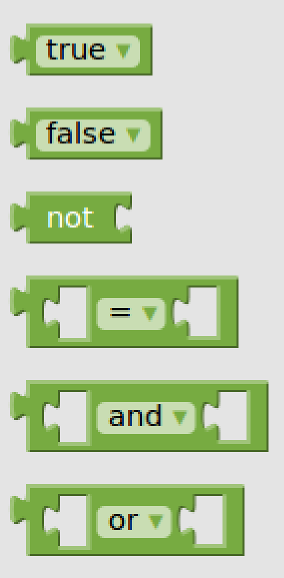
## 6.- Operacions Aritmètiques

Són les que es fan amb valors numèrics. Sumes, restes, multiplicacions, divisions, potències, etc.



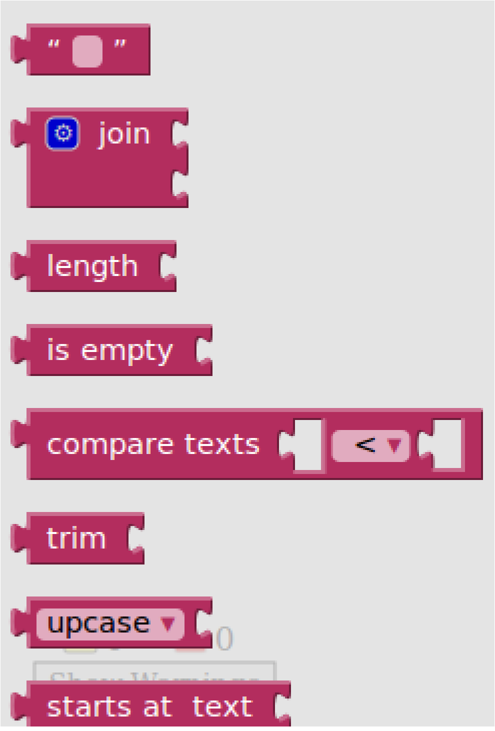

## 7.- Operacions lògiques

Són les que donen com a resultat els valors "Si" o "No", True o False, 1 o 0, etc. Poden ser el resultat d'una comparació i/o una operació conjuntiva, disjuntiva o negació (and, or o not).



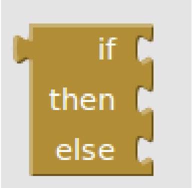

## 8.- Operacions amb cadenes alfanumèriques

Aquestes són les que s'usen per manipular cadenes de text. Entre elles podem tenir les concatenacions de cadenes (join), determinar-ne la longitud, transformar a majúscules, a minúscules, etc.



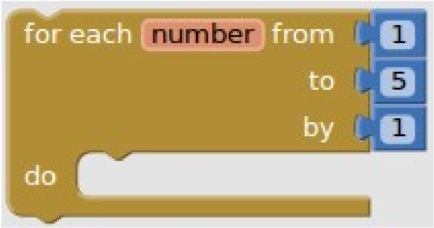
## 9.- Estructures condicionals

Son les estructures de control de flux del programa que permeten executar trossos de codi condicionalment al resultat de l'avaluació d'una condició que retorna un valor lògic.

| | | |
|---|---|---|
| | |  |
| Aquesta és l'expressió més simple, executarà les ordres indicades a **then** solament si la condició indicada a **if** és certa. | Aquesta expressió més complexa i derivada de l'anterior, permet múltiples condicions i, en conseqüència, múltiples ordres a executar, indicades a cada **then**, segons sigui la condició avaluada com a certa a cada **if**. | Aquesta estructura és per efectuar assignacions de dades a variables de forma condicional. |

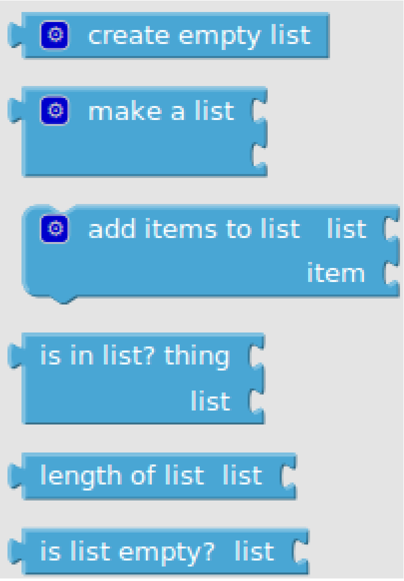
## 10.- Estructures iteratives

Aquestes són les que permeten repetir un conjunt d'instruccions múltiples vegades fins ( o mentre) es compleix una determinada condició.

| | |
|---|---|
| |  |
| L'estructura **while** repeteix les instruccions indicades per **do** mentre sigui certa la condició lògica indicada en el **test**.<br>El nombre de repeticions no es conegut en el moment de iniciar la iteració. | L'estructura **for** repeteix les instruccions indicades per **do** per a cada valor numèric comptat entre el valor de **from** i el valor de **to**, incrementant segons s'indica a **by**.<br>El nombre de repeticions es conegut en el moment d'iniciar la iteració. |

## 11.- Llistes

Les llistes són un conjunt de dades indexades i enumerades des de 1 fins a n. Tot el conjunt d'elements de la llista es tractat com una sola entitat referida pel mateix nom de la llista i cada element de la llista es referència amb el seu índex.



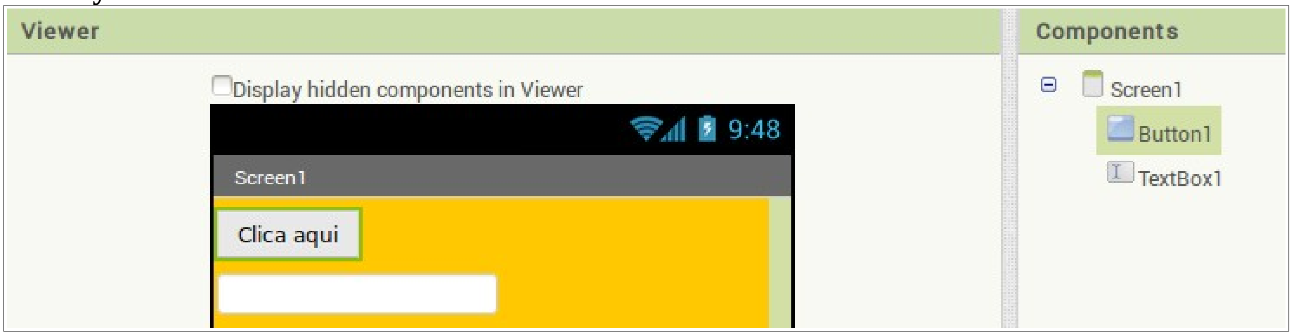

## 12.- Estructures de tractament de llistes

Aquestes estructures poden executar un conjunt d'ordres per a cadascun dels elements (items) que formen la llista.

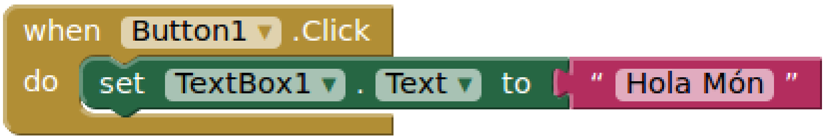
## 13.- Pràctica no. 1. Hola món.

Programa inicial molt simple que escriu el text “Hola món” en el quadre de text quan es clica sobre el botó.

**Disseny:**



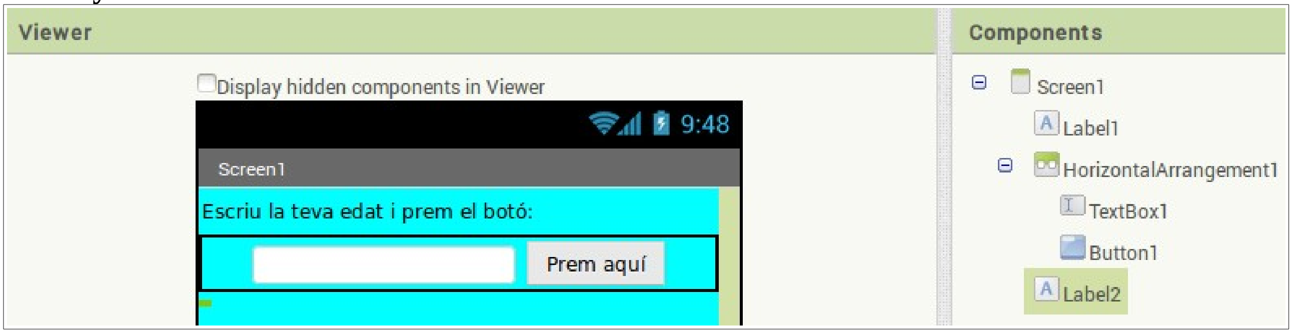

**Blocs:**



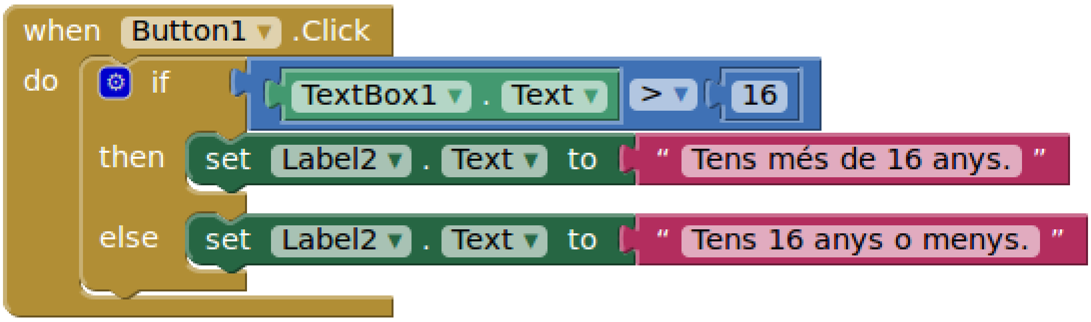
## 14.- Pràctica no. 2. Compara edat.

Donada una edat, determina si és més gran o menor que 16 anys. S'utilitza una estructura condicional If-then-else per comparar l'edat introduïda amb el 16 i decidir el text a mostrar en el quadre de text.

**Disseny:**



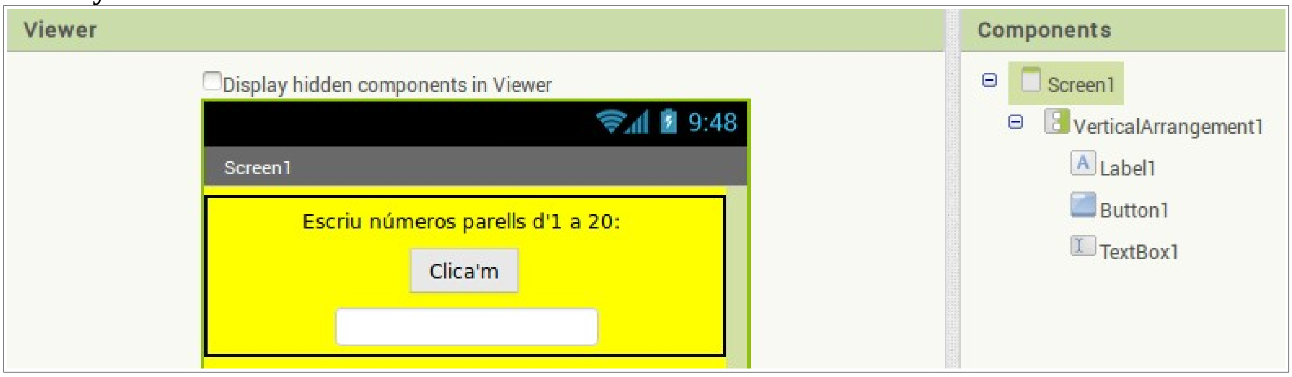

**Blocs:**



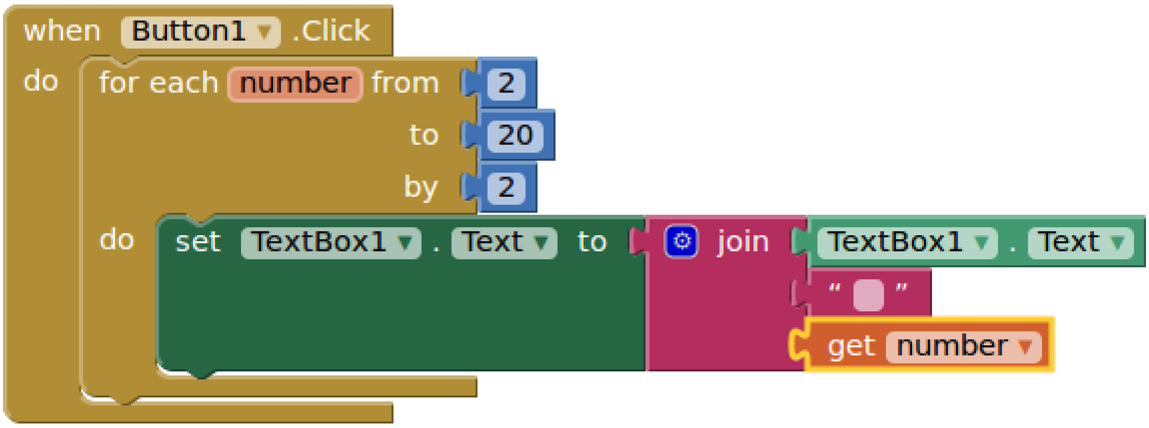

## 15.- Pràctica no. 3. Llista nombres parells.

Programa que escriu la llista de números parells entre 1 i 20. S'utilitza una estructura iterativa **for** que, a cada repetició, incrementarà el valor de la variable comptadora *number* en dos unitats començant des de 2 fins a 20.
A cada iteració s'acumula a la propietat *Text* de *TextBox2* el nou valor de *number*, concatenat amb el contingut anterior de Text i un espai en blanc « « mitjançant **join**.

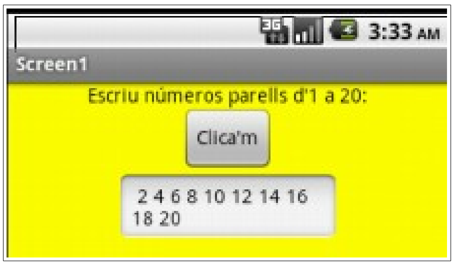
**Disseny:**



**Blocs:**



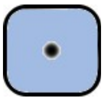
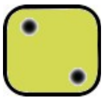
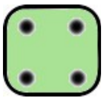
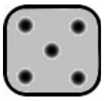
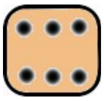

**Resultat:**



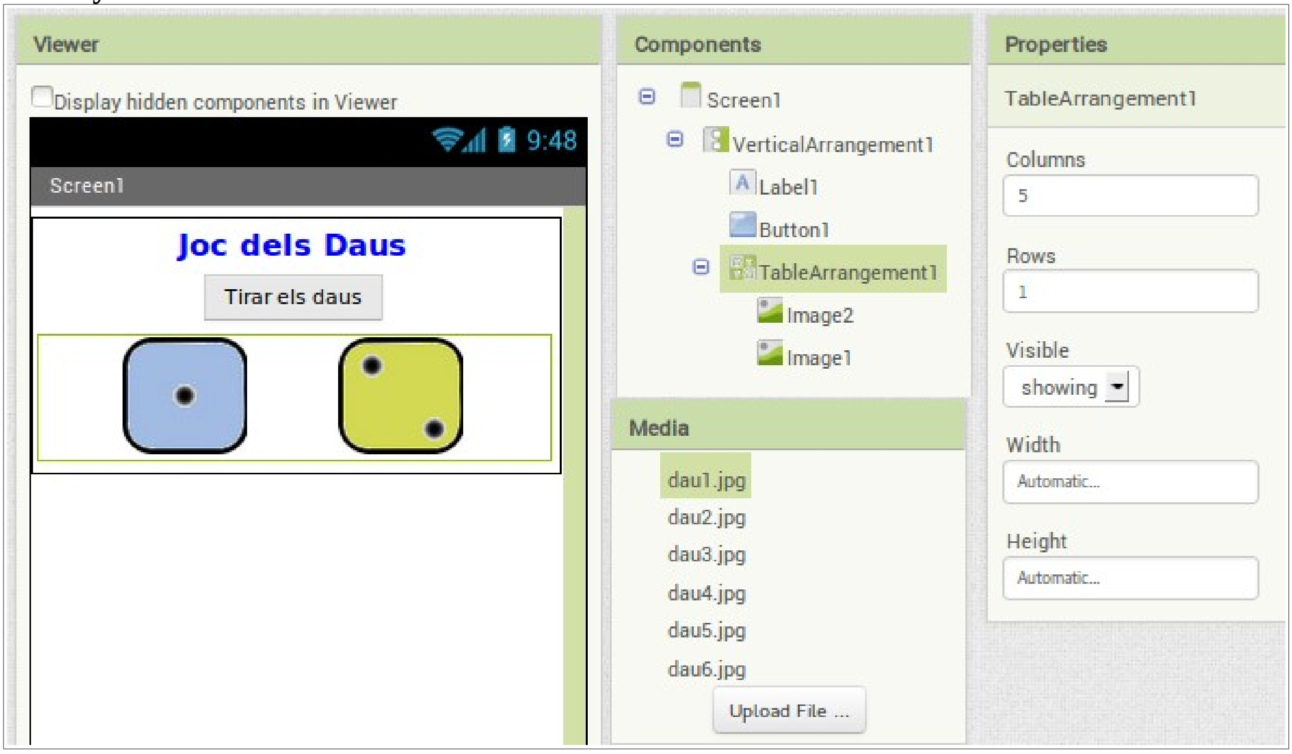

## 16.- Pràctica no. 4. Daus.

Crear un programa que en clicar sobre un boto faci una "tirada" de dos daus i en mostri el resultat obtingut aleatòriament.

Carreguem al projecte els dibuixos, semblants als següents, corresponents a les 6 cares del dau fetes amb un programa d'edició d'imatges:

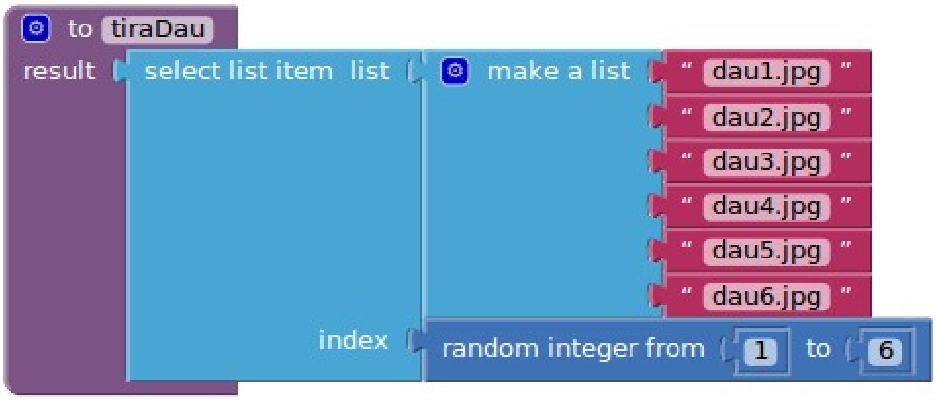
| | | | | | |
|---|---|---|---|---|---|
|  dau1.jpg |  dau2.jpg | dau3.jpg |  dau4.jpg |  dau5.jpg |  dau6.jpg |

**Disseny:**

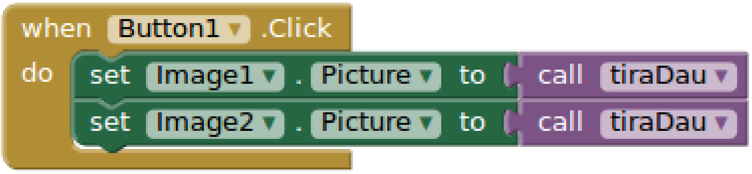



**Blocs:**

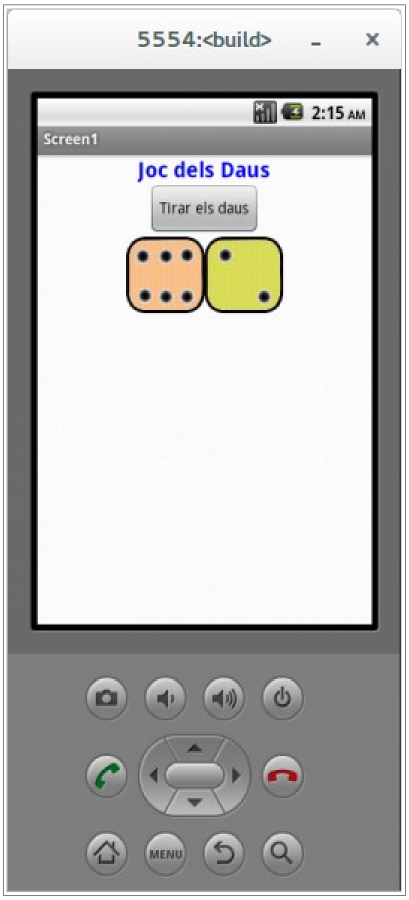
Crearem el següent procediment que, en invocar-lo, retorna un nom de fitxer corresponent a una imatge d'una cara del dau obtinguda aleatòriament de la llista de noms de fitxers.




El programa principal, executat en clicar sobre el botó “Tirar els daus”, elegirà, per a cada imatge, un fitxer elegit aleatòriament en invocar el procediment “tiraDau” que s'ha definit prèviament.



**Resultat:**



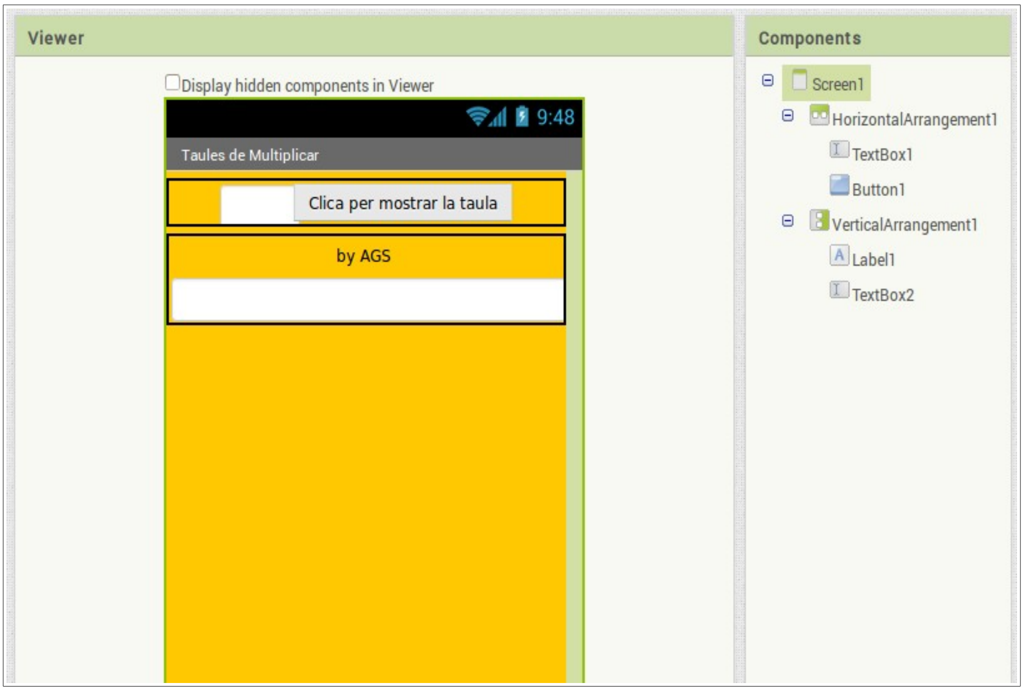

## 17.- Pràctica no. 5. Taules de multiplicar.

Programa que mostra la taula de multiplicar d'un número introduït per l'usuari. Utilitzem una variable acumuladora **t** per crear el text de la taula que es mostra al final de l'execució. Per fer el salt de línia, utilitzem el caràcter especial: **n**.

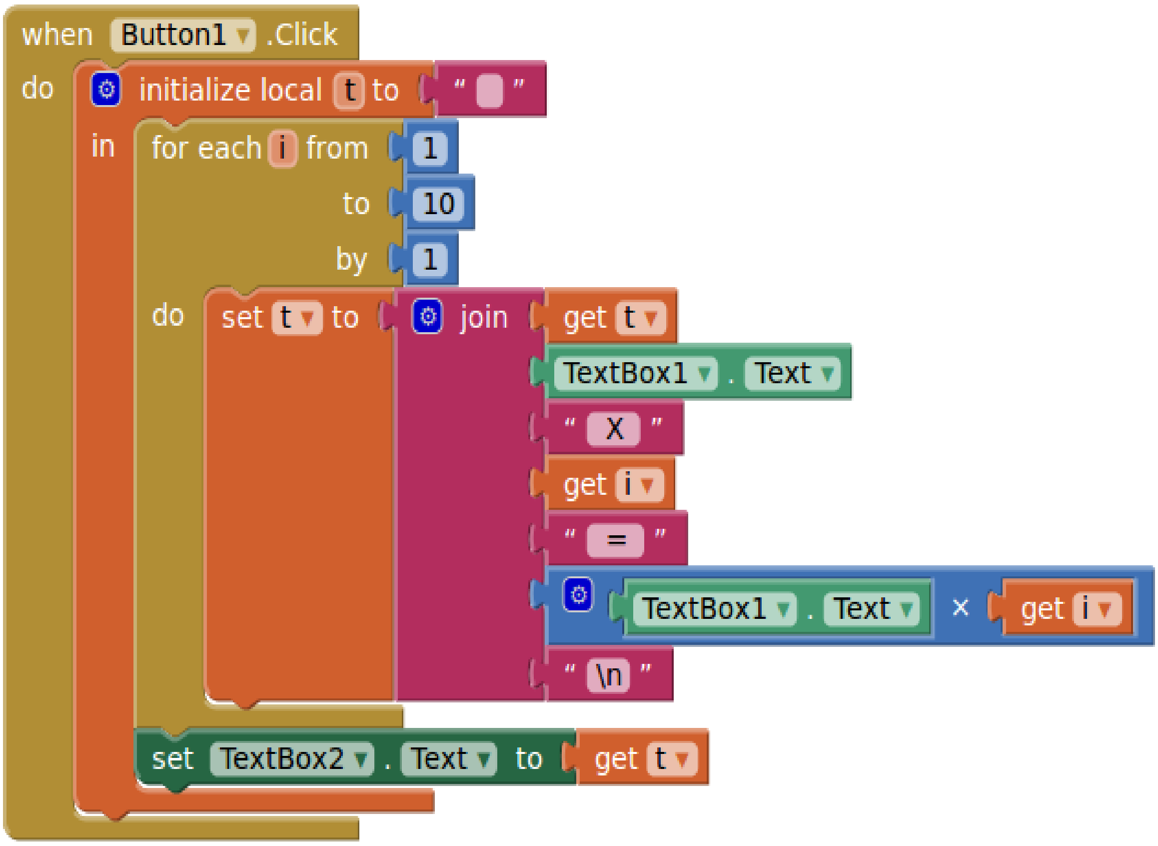
Mitjançant el **for** iterem 10 vegades i, a cada iteració, la variable comptadora **i** valdrà el valor que utilitzem per multiplicar el valor de cada fila a la taula. A cada iteració, acumulem (concatenant, join) a la variable acumuladora **t,** el text corresponent creat. Finalment, després de les 10 iteracions es mostra a la propietat Text de TextBox2 el desenvolupament de la taula de multiplicar elegida.

En el TextBox2, per mostrar múltiples files cal activar-li la propietat **Multiline** i, si volem evitar que es pugui modificar el seu contingut caldrà desactivar la propietat **Enabled**.

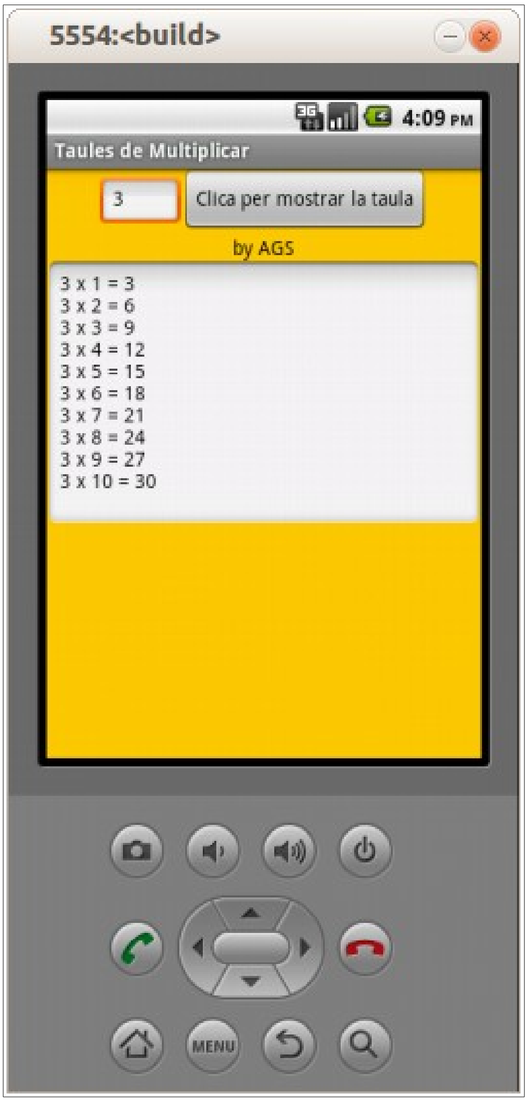
## Disseny:




## Blocs:



**Resultat:**

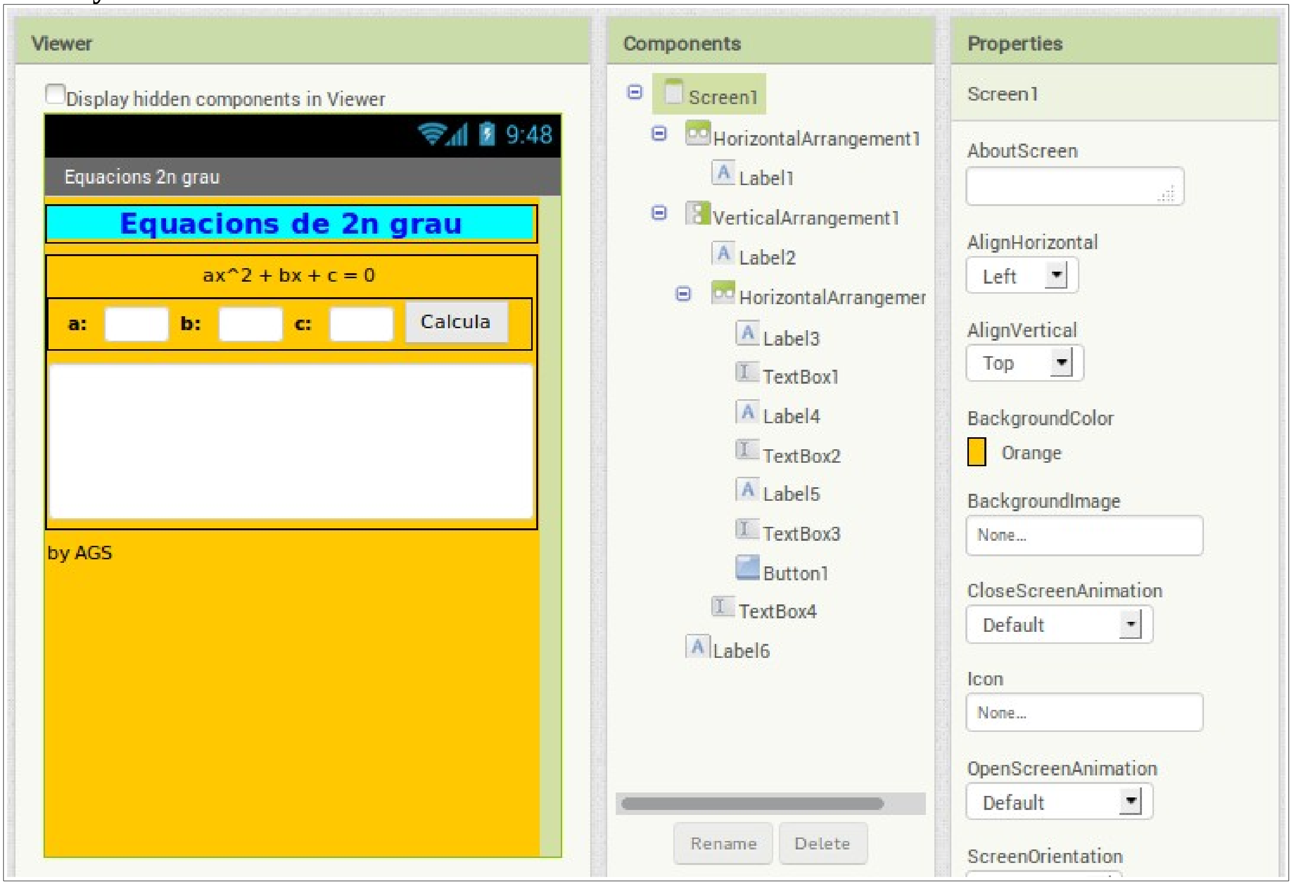



## 18.- Pràctica no. 6. Equacions de $2^n$ grau.

Programa que serveix per calcular les arrels d'una equació de segon grau en la forma: $ax^2 + bx + c = 0$, aplicant l'equació:

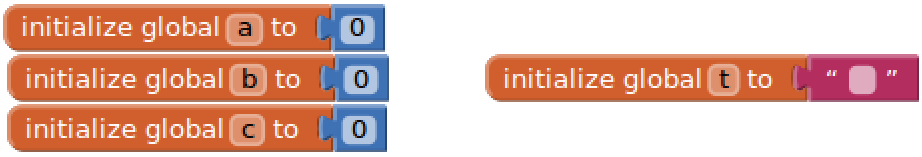
$$x=\frac{-b\pm\sqrt{b^2-4\cdot a\cdot c}}{2\cdot a}$$

## Disseny:

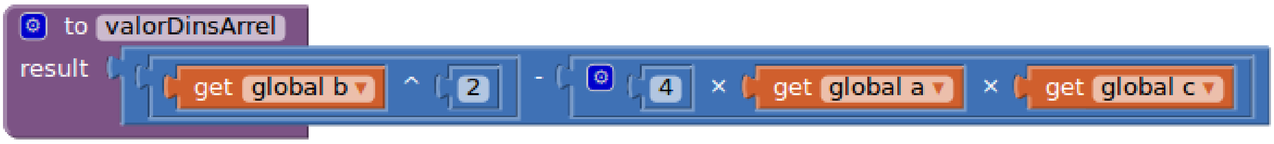



## Blocs:

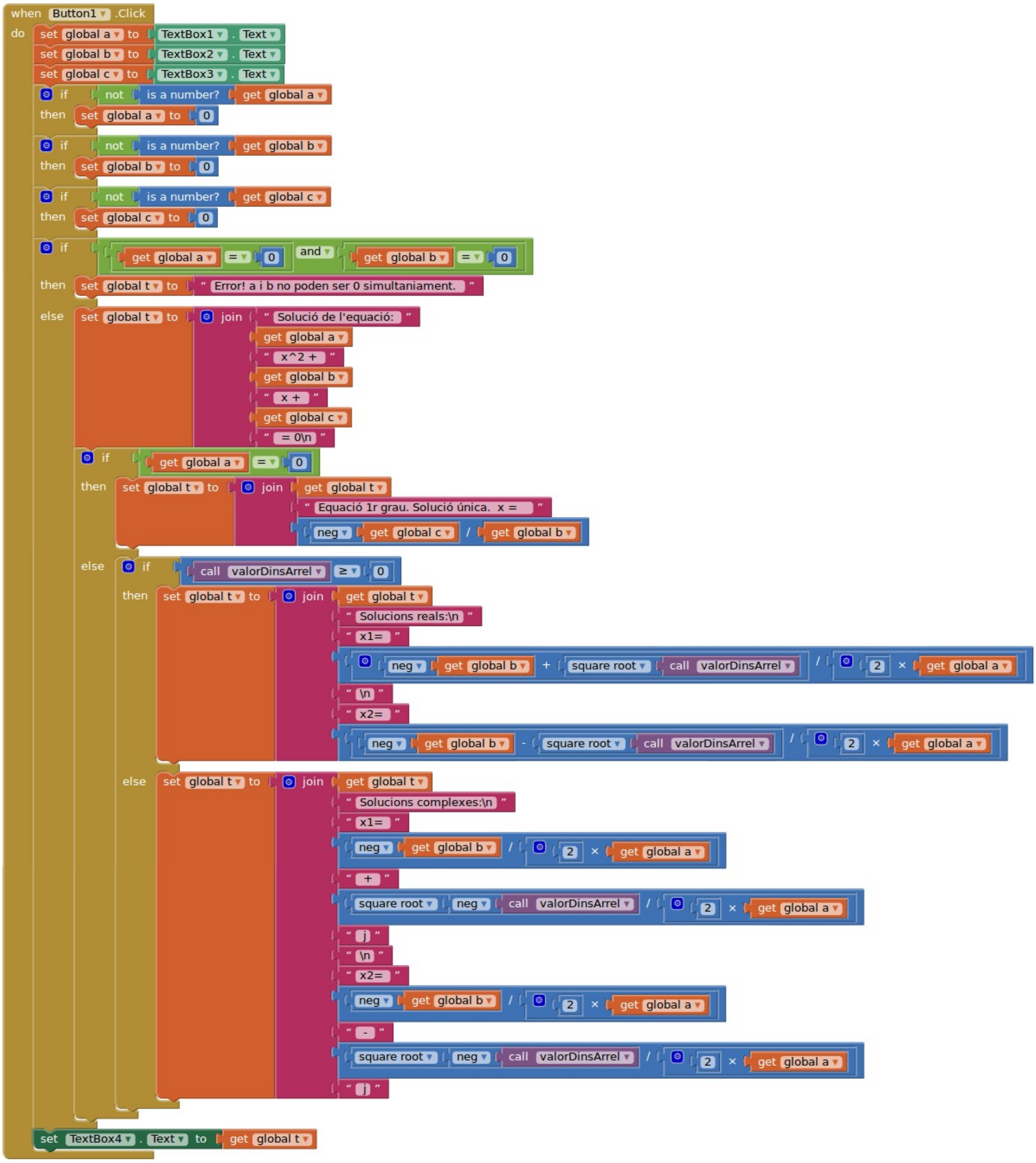
Declarem les variables globals inicialitzades al valor numèric 0 (**a**, **b**, **c**) i la variable global **t** inicialitzada a un valor textual buit. Les variables **a**, **b** i **c** contindran els coeficients de l'equació a resoldre i **t** serà una variable acumuladora que acumularà el text que es mostrarà al final com a resultat.




Declarem també un procediment de nom "valorDinsArrel" que calcularà el valor numèric de dins de l'arrel de l'equació ( $b^2 - 4 \cdot a \cdot c$ ). Aquest procediment s'utilitzarà diverses vegades dins el programa principal.



Programa principal que s'executa en el moment de clicar sobre el botó “Calcula”.



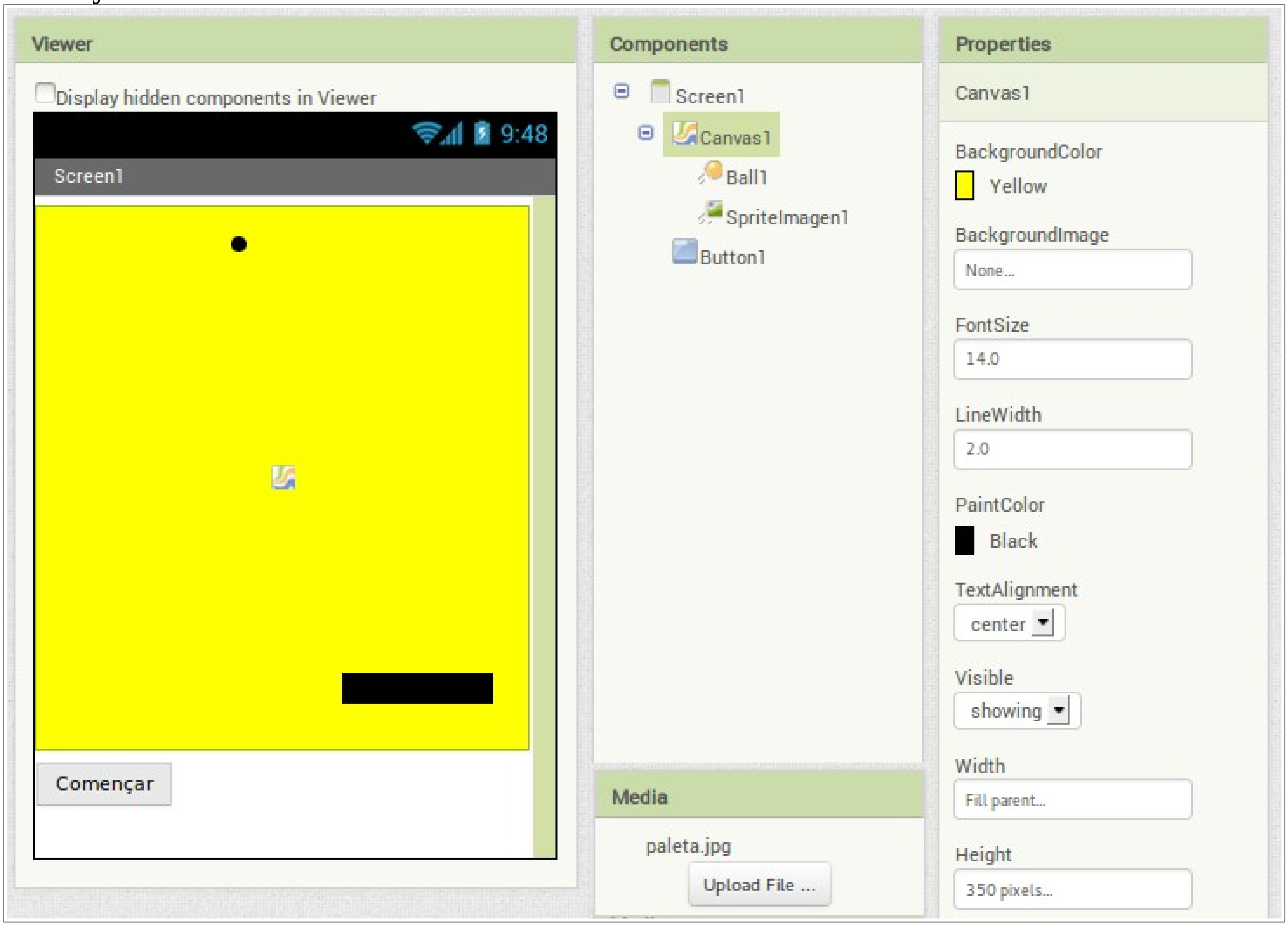
## Resultat:

5554:<build>
11:50 PM
Equacions 2n grau
Equacions de 2n grau
ax^2 + bx + c = 0
a: -11 b: 4 c: 7 Calcula
Solució de l'equació: -11 x^2 + 4 x + 7 = 0
Solucions reals:
x1= -0.63636
x2= 1.0
by AGS
MENU

# 19.- Pràctica no. 7. Ping-pong.

Programa que implementa el joc del ping-pong sobre la pantalla. Utilitzem un Canvas on es desenvoluparà tota l'activitat.

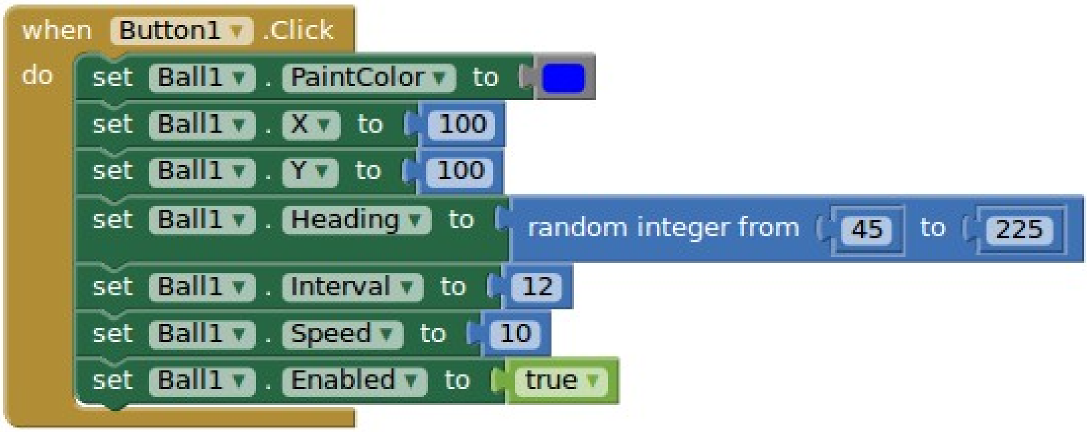
**Disseny:**



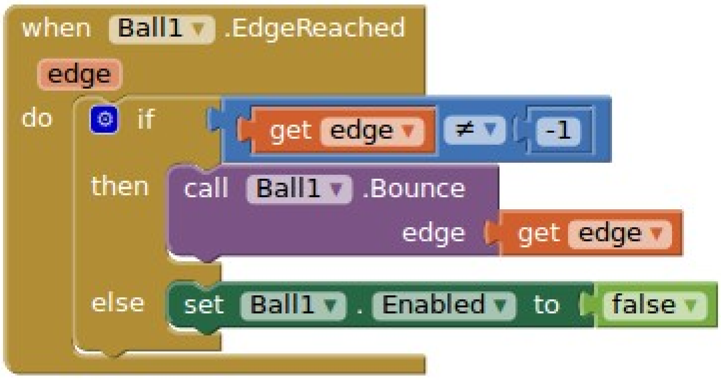

Amb un editor gràfic, crearem una imatge de nom “paleta.jpg” similar a la següent: , carregarem aquesta imatge i l'assignarem a la propietat Picture de “SpriteImagen1”. Ubicarem el “SpriteImagen1” a les coordenades: x:200px i y:300px

**Blocs:**

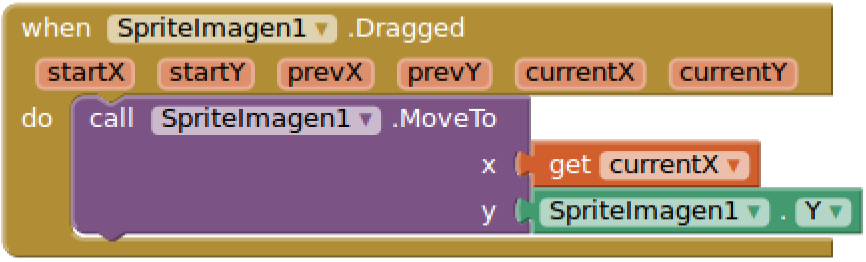
Amb el botó “Començar” inicialitzem els principals paràmetres de la pilota que es mourà sobre la pantalla.




Quan la pilota toca un costat del canvas, excepte el de baix, rebota en sentit contrari. En canvi, quan toca el costat de baix, s'atura i acaba el joc fins que es torni a reinicialitzar amb el botó.

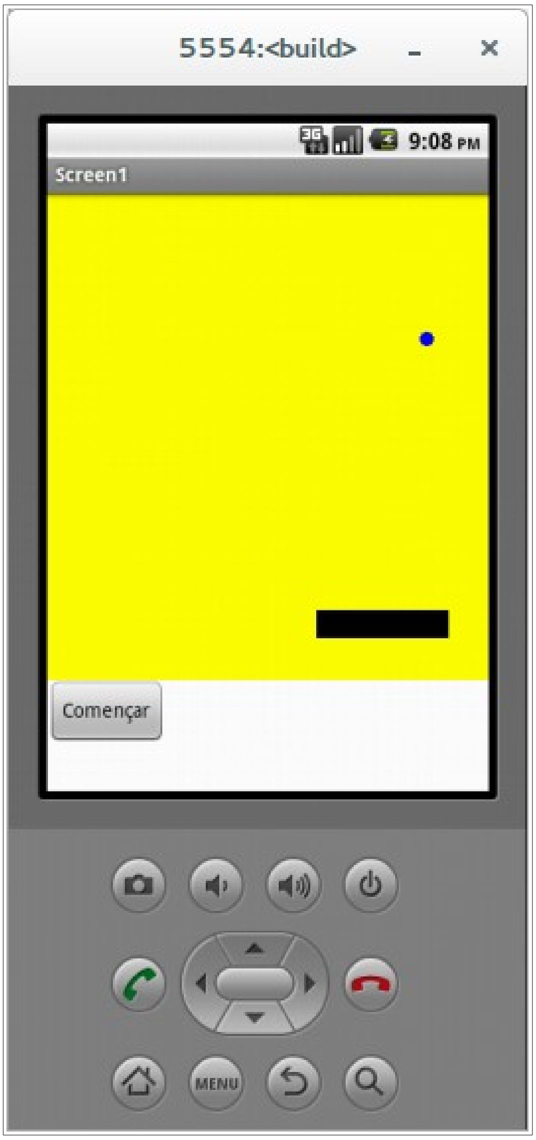



Quan arrastrem la paleta amb el dit, solament permetrà desplaçament horitzontal




Quan la pilota xoca amb qualsevol altre objecte dins el canvas (la paleta), aquesta rebotarà en una direcció en angle complementari al que tenia abans del xoc.

**Resultat:**

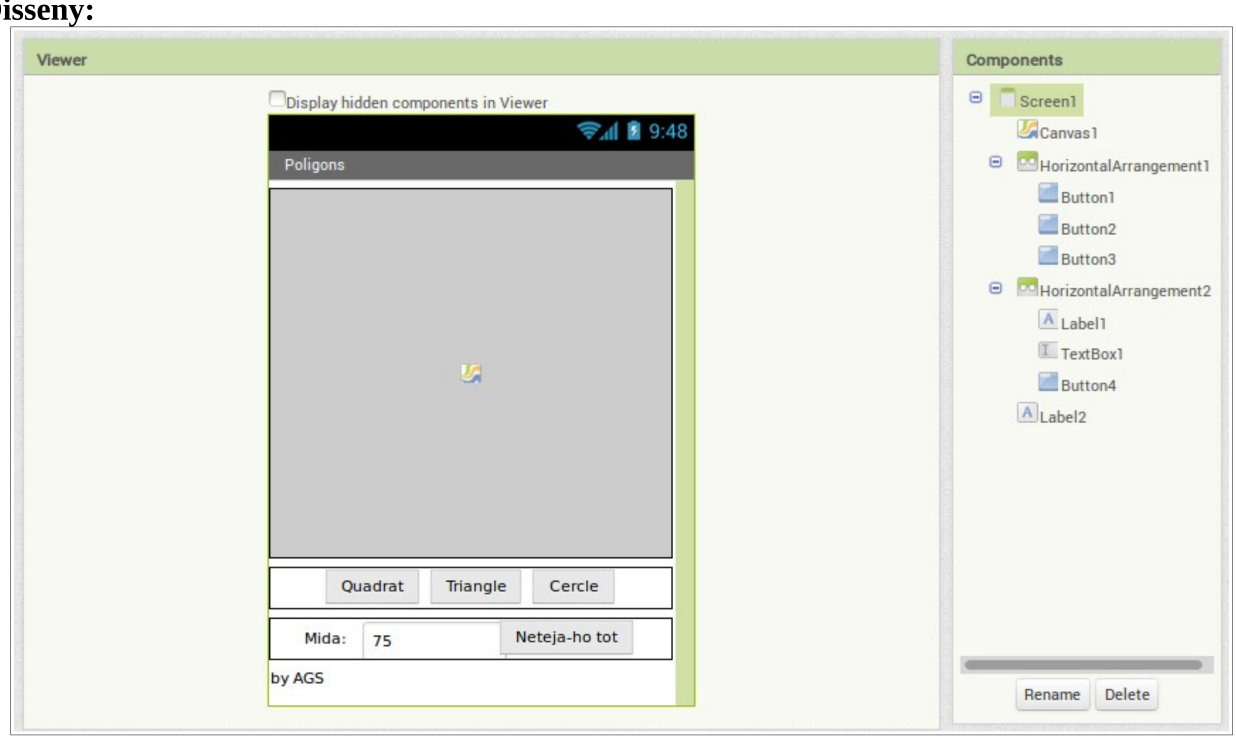



## 20.- Pràctica no. 8. Polígons.

Programa que dibuixa un quadrat, un triangle o un cercle sobre la pantalla, amb les dimensions i, a partir del punt indicat per nosaltres.

Per això s'utilitza una àrea de dibuix (*Canvas1*) i sobre ella es faran les representacions gràfiques amb les opcions de representació de línies, cercles, punts, etc. disponibles per al canvas.
A *TextBox1* li activem la propietat *NumberOnly* per garantir que sempre tindrà un valor numèric.

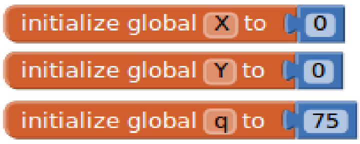
## Disseny:




## Blocs:

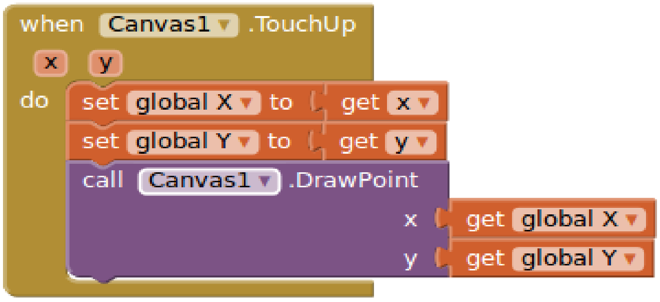
Inicialitzem les variables globals X, Y i q, les dos primeres indicaran les coordenades sobre el canvas i la q indica la longitud del segment a representar en el quadrat o triangle o el radi en el cercle.

Cal tenir en compte que les coordenades X i Y sobre el canvas s'inicien amb el punt (0,0) a la cantonada superior esquerra i augmenten positivament cap a la dreta i cap avall respectivament per a *x* i *y*.

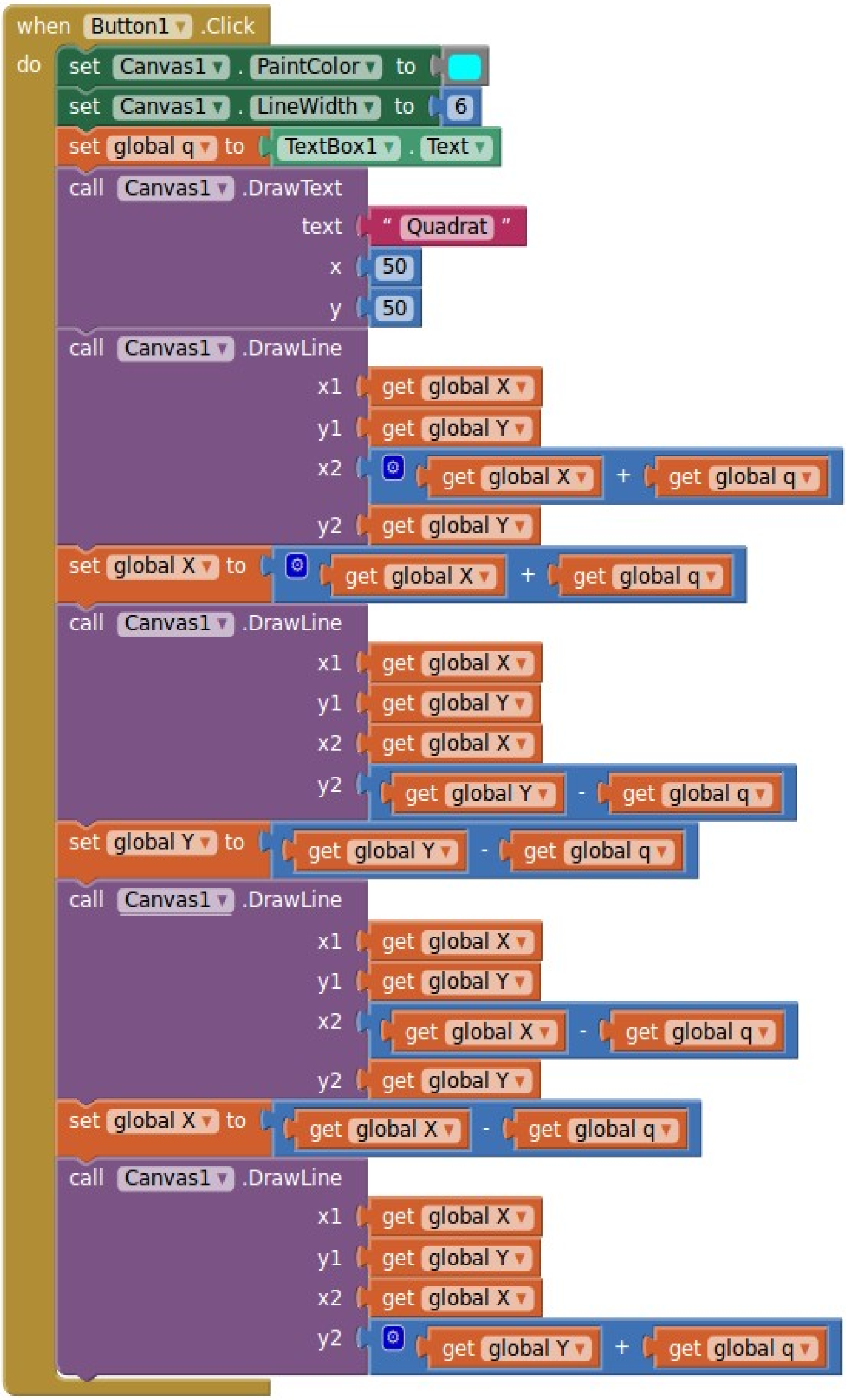



Per cada representació, la variable *q* serà la mida dels costats o el radi dels polígons i es prendrà des del valor indicat en el quadre de text *TextBox1*.

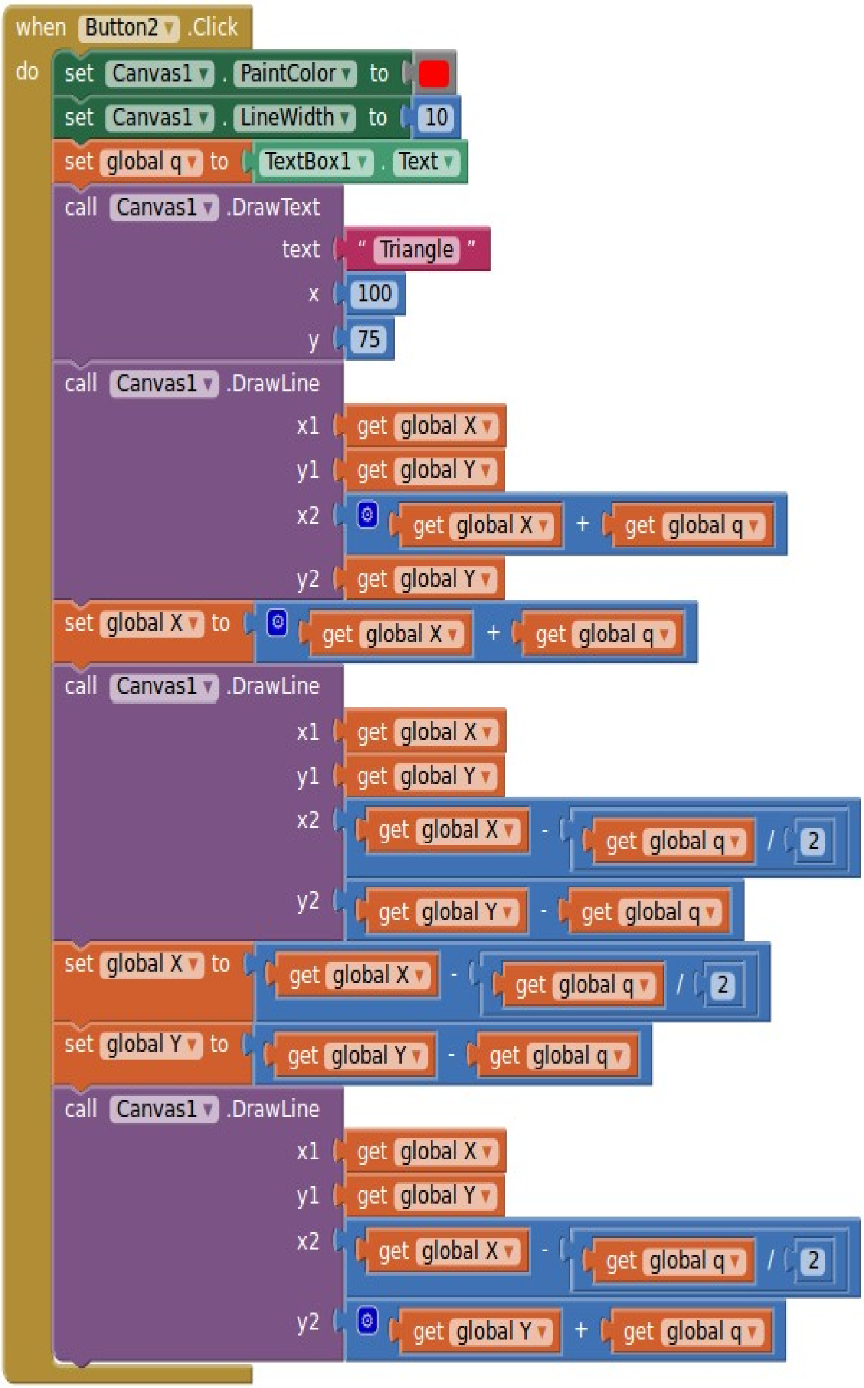
En prémer sobre la superfície del *Canvas1*, definim un punt, que desarà les coordenades a les variables globals X i Y, on s'iniciarà la representació del polígon, seleccionat posteriorment.



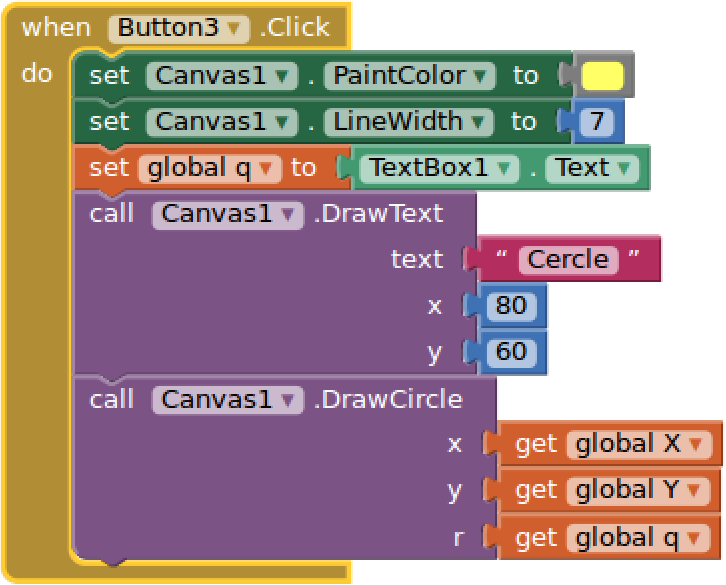
En clicar sobre el botó “Quadrat” s'inicia la representació del quadrat mitjançant quatre segments de línia de longitud *q* a partir del punt X i Y:



En clicar sobre el botó “Triangle” s'inicia la representació del triangle mitjançant tres segments de línia a partir del punt indicat a X i Y:

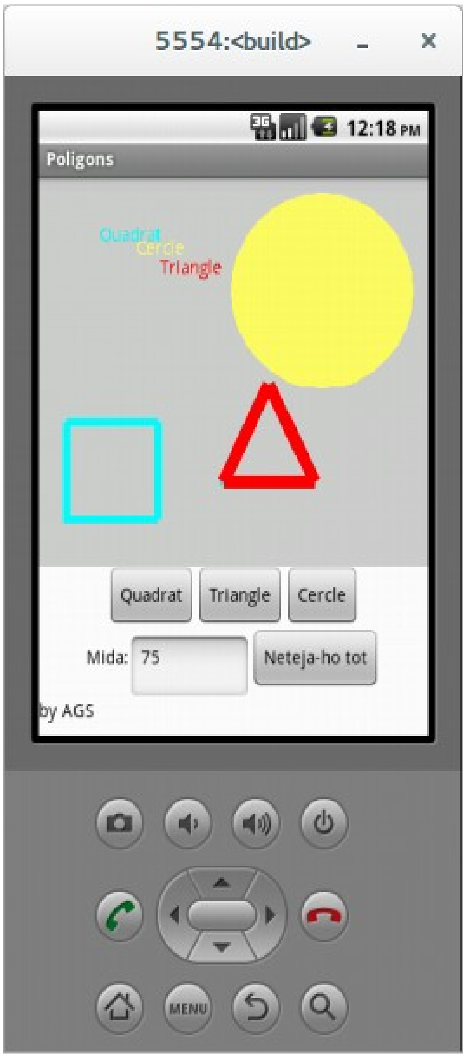


En clicar sobre el botó "Cercle" s'inicia la representació del cercle que té com a centre les coordenades X i Y i radi q:




Per netejar l'àrea de dibuix (el canvas) cliquem sobre el botó "Neteja-ho":

## Resultat:


5554:<build>
12:18 PM
Poligons
Quadrat
Cercle
Triangle
Quadrat
Triangle
Cercle
Mida:
75
Neteja-ho tot
by AGS
MENU

## 21.- Exercicis

**Exercici 1:**
Crea un programa que, havent introduït una frase, en retorni el següent:
- La mateixa frase, repetida, en majúscules i minúscules.
- La longitud de la frase.
- La frase al revés.

**Exercici 2:**
Crea un programa de càlculs estadístics que sol·liciti un nombre indeterminat de números enters i, un cop introduïts, informi de la llista dels números introduïts, en ordre d'introducció i en ordre ascendent, la quantitat de números, la seva suma, la mitjana, la mediana i la moda.

**Exercici 3:**
Crear un programa que implementi dos sprites que estiguin en moviment aleatori constant sobre la pantalla i, quan xoquin entre ells dos, s'aturin i mostrin, amb un notificador, les coordenades X,Y del lloc de col·lisió. També, amb un cronòmetre, s'ha de mesurar el temps que es triga en produir-se el xoc.

**Exercici 4:**
Crear un programa que, en prémer un botó, mostri sobre la pantalla la data i hora, les coordenades geogràfiques de la ubicació actual i la altitud.
A més, aquestes dades s'han de desar de forma permanent, és a dir, en aturar l'aplicació i tornar-la a iniciar s'ha de conservar les dades desades prèviament. En prémer un altre botó, ens ha de mostrar tota la llista de dates i hores i coordenades geogràfiques que han estat desades prèviament. Amb un altre botó hem de eliminar totes les dades que han estat desades de forma permanent.

**Exercici 5:**
Crear un programa que ens permeti elegir un fitxer de música des d'una llista o des de l'emmagatzematge del dispositiu i posteriorment el reprodueixi.

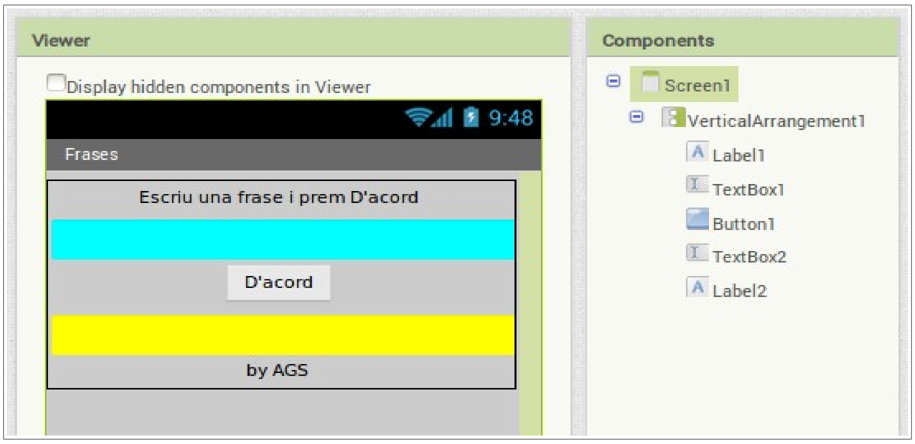
---------------------------------------------------------

**Solucions:**

| | | | | |
|---|---|---|---|---|
| Exercici1_Frases.apk | Exercici2_Estadist.apk | Exercici3_Xocs.apk | Exercici4_Coordenades.apk | Exercici5_Player.apk |

## Exercici 1. Solució:

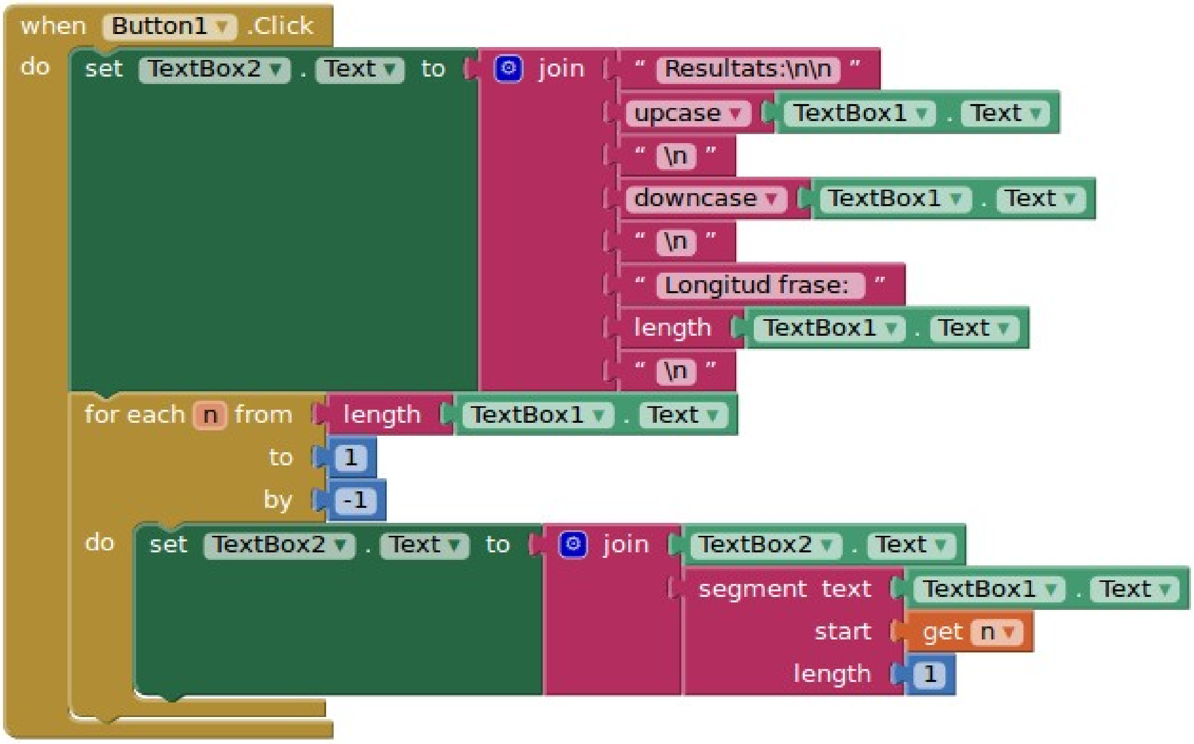
### Disseny:



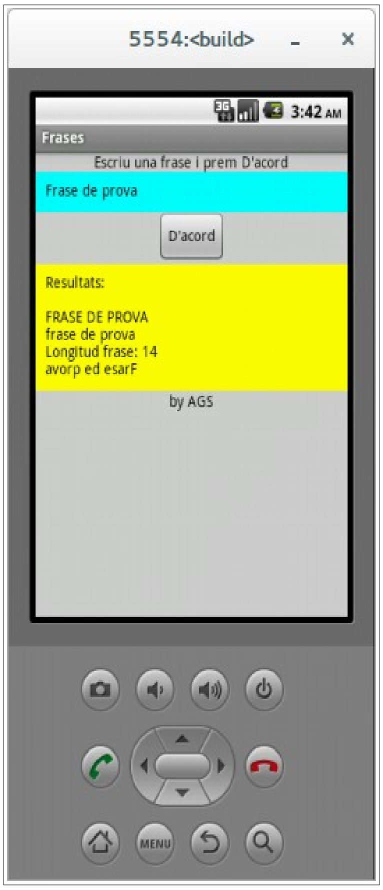

El *TextBox2* ha de tenir la propietat *Multiline* activada, i si no es vol que es pugui modificar el contingut resultant, també es pot desactivar la propietat *Enabled*

### Blocs:

Introduïm **\n** per que el text es vegi amb línies diferents (sempre que el quadre de text tingui activada la propietat *Multiline*).



## Resultat:




## Exercici 2. Solució:

## Disseny:

El *TextBox1* ha de tenir activada la propietat *NumberOnly* i el *TextBox2* ha de tenir activada la propietat *Multiline.*

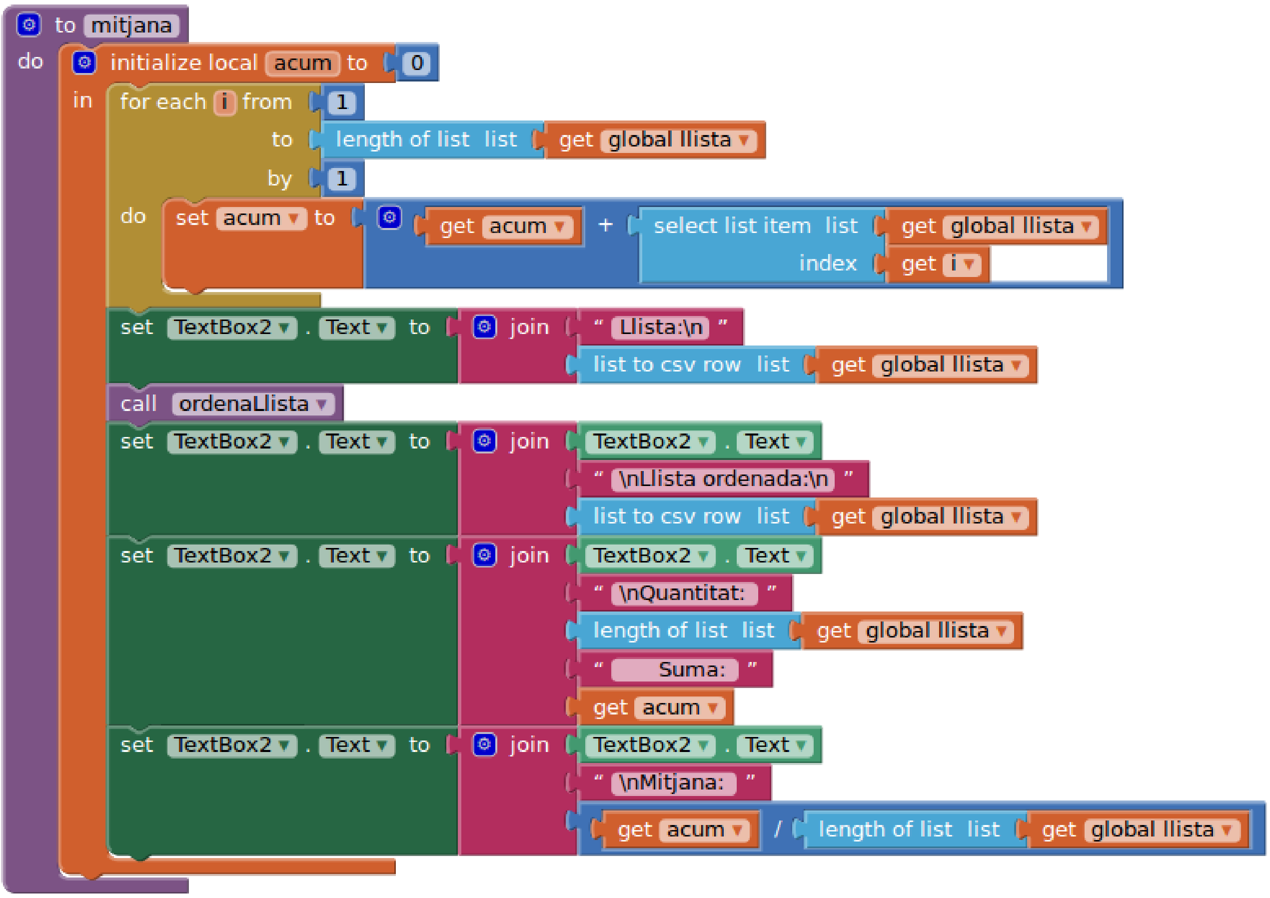
**Blocs:**

Inicialitzem la variable global *llista* que contindrà els valors de la llista de dades i serà utilitzada en els diferents mòduls i funcions del programa:

El programa principal s'invocarà en clicar sobre el botó “Afegir”. Si s'afegeix un valor numèric, aquest s'afegirà a la llista, en canvi, si s'afegeix un punt “.” es faran els calculs per mostrar les dades estadístiques sol·licitades. Per calcular les dades estadístiques s'ha fet trossejant el programa amb els procediments “mitjana”, “mediana” i “freqüències” que es descriuen posteriorment:

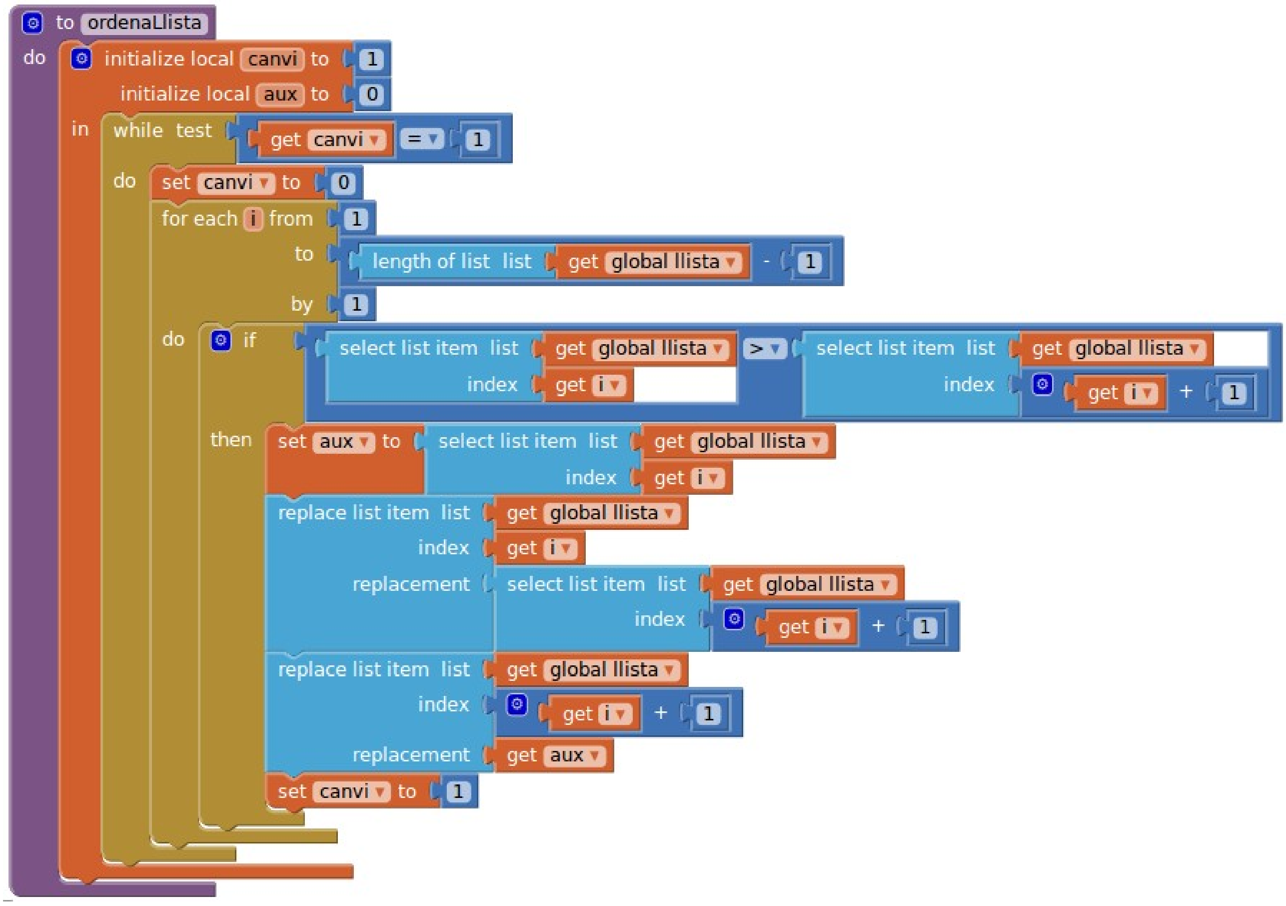
Per buidar la llista i començar de nou, cliquem sobre el botó “Buida llista”:

El procediment "mitjana" mostrarà la llista introduïda, la llista ordenada, el nombre de items de la llista, la seva suma i la seva mitjana.



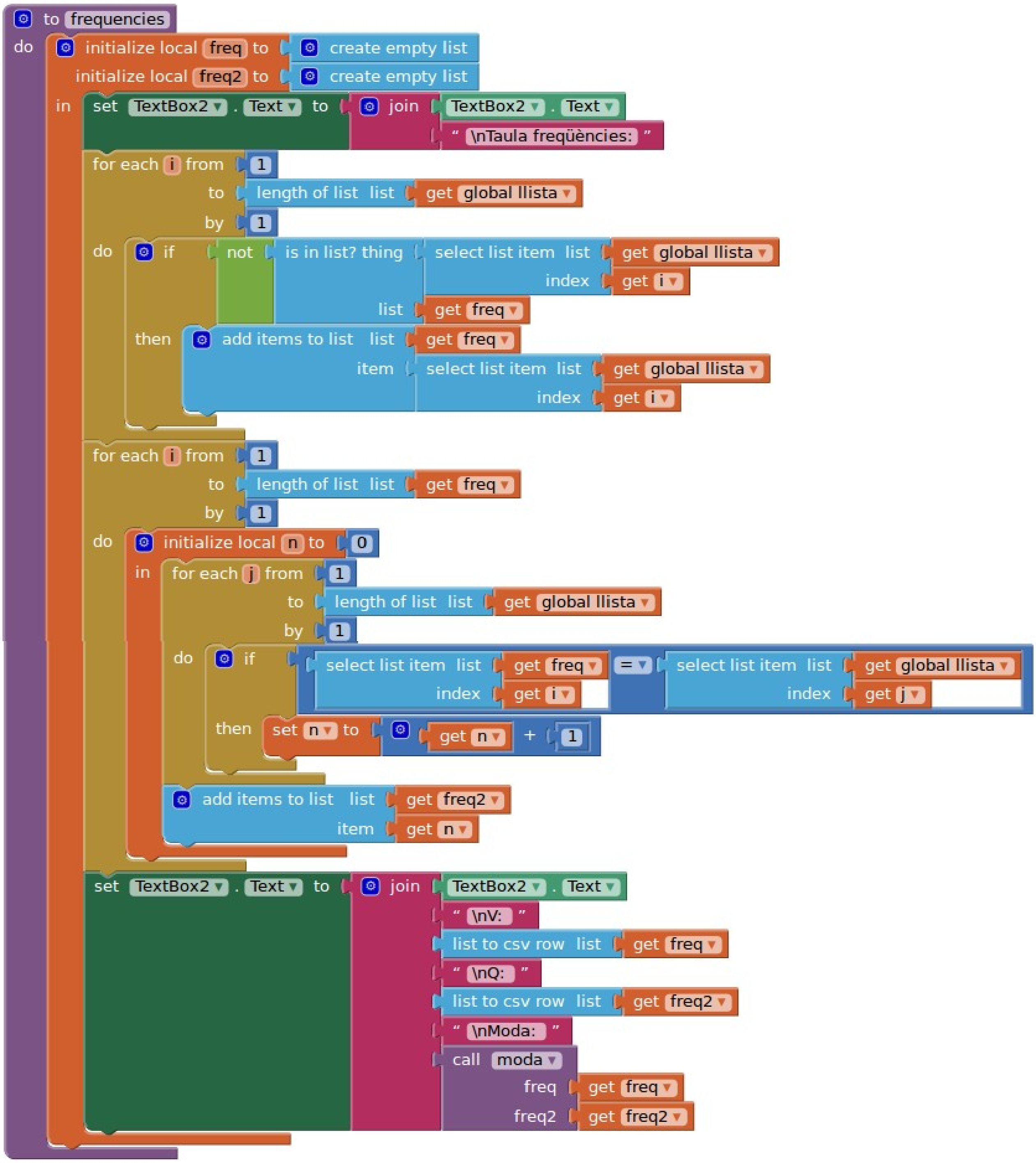

El procediment *mitjana* invoca al procediment *ordenaLlista*.

El procediment *ordenaLlista* és per fer la ordenació d'una llista numèrica global de nom *llista*, en ordre ascendent, utilitzant el mètode de la bombolla.



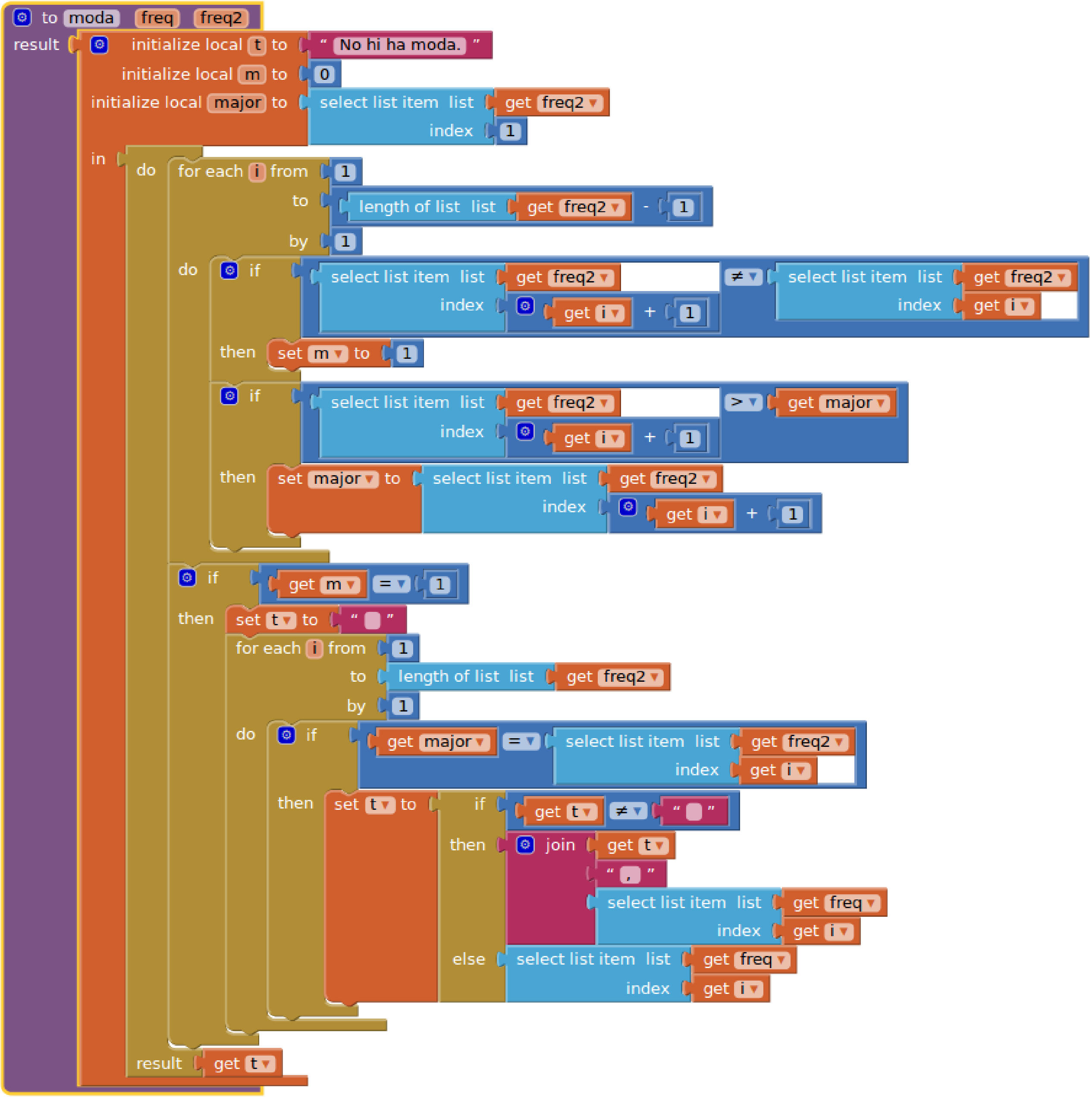

El procediment *mediana* s'encarrega de calcular la mediana de la llista ordenada prèviament. El valor de la mediana es determina de forma diferent segons si el nombre de termes de la llista és parell o senar.

El procediment *freqüències* s'encarrega de formar la taula de freqüències, aquesta taula es necessària per determinar, posteriorment, la moda. La taula de freqüències (formada per cada valor de la llista i el nombre de vegades que es repeteix) esta implementada en dos llistes locals: *freq* i *freq2*, la primera conté els valors de la llista i la segona la quantitat de vegades que es repeteix, la correspondència entre les dos taules és la posició indicada per l'índex de cada taula que ha de ser el mateix per cada parella valor-freqüència. Aquest procediment, un cop ha determinat la taula de freqüències, invocarà al procediment *moda* transferint-li les dos llistes indicades.




Aquest darrer procediment, *freqüències,* també invoca a un altre procediment anomenat: *moda* (call moda) passant-li com a paràmetres les llistes *freq* i *freq2* que formen la taula de freqüències. El procediment *moda* rep com a paràmetres les llistes *freq* i *freq2* i retorna el valor de la moda.

El procediment *moda* és l'encarregat de cercar el valor, o valors, amb més freqüència, i, si tots tenen la mateixa freqüència, indicar que no existeix moda, el valor resultant es retorna a l'element invocant (call moda) dins el procediment *freqüències*.

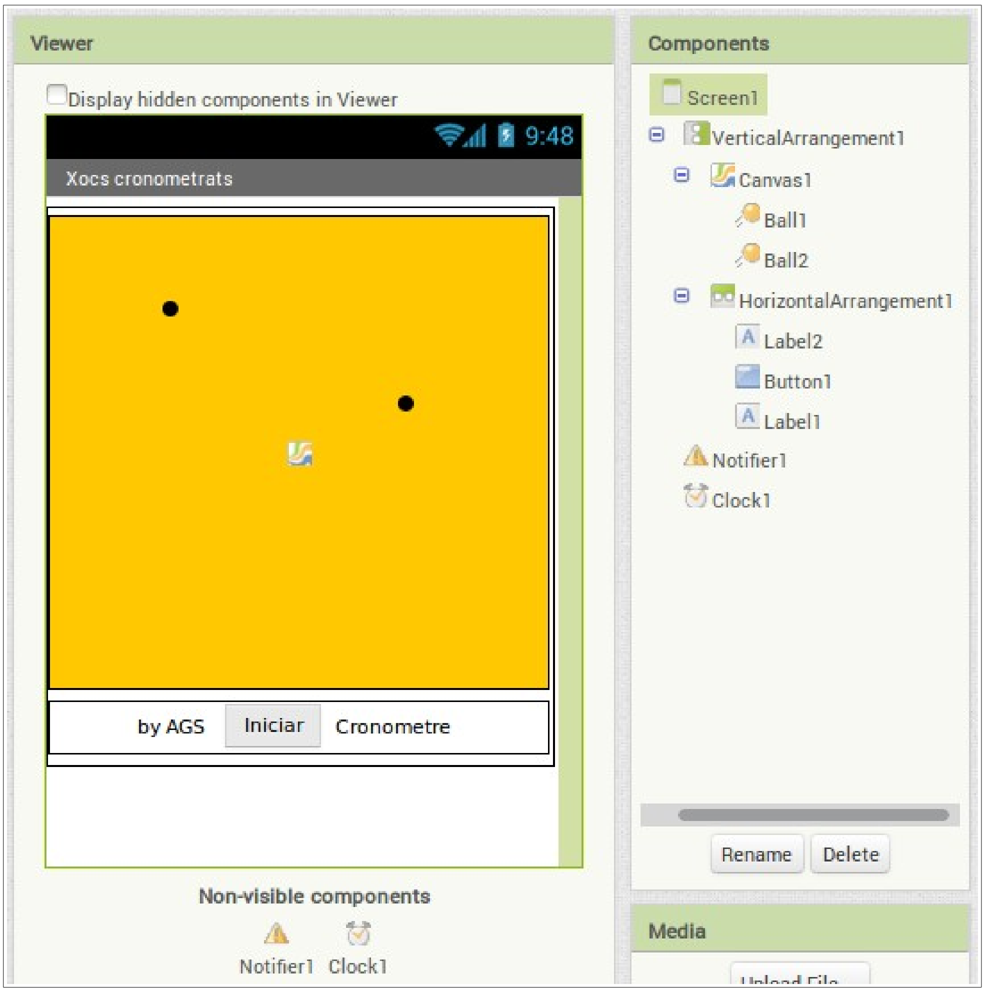


## Resultat:

5554:<build>
5:03 AM
Estadístiques
Afegeix nos. enters a la llista.
Insereix un . per acabar la llista i mostrar resultats
Afegir
Buida llista
Llista:
"2","5","4","7","5","8","9","8"
Llista ordenada:
"2","4","5","5","7","8","8","9"
Quantitat: 8 Suma: 48
Mitjana: 6
Mediana: 6
Taula freqüències:
V: "2","4","5","7","8","9"
Q: "1","1","2","1","2","1"
Moda: 5, 8
by AGS
MENU

## Exercici 3. Solució:

### Disseny:

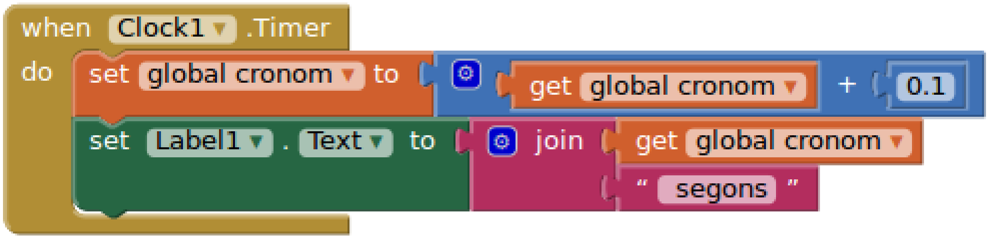



Utilitzem els component ocults *Notifier1* per mostrar sobre pantalla el text informatiu de les coordenades de col·lisió i el *Clock1* per generar un esdeveniment cada dècima de segon que s'encarrega d'incrementar el cronòmetre en dècimes de segon.

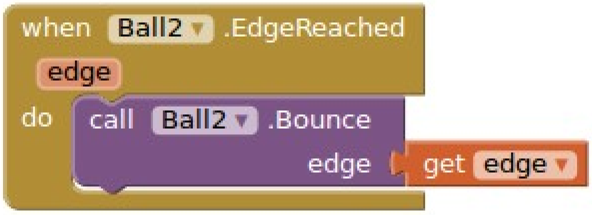
### Blocs:

Inicialitzem variables globals i definim, per quan es clica el botó de Iniciar, els valors inicials de les propietats corresponents a color, posició X i Y, direcció, velocitat (pas i temps) de forma aleatòria per a cadascuna de les dos boles que es mouran sobre la pantalla. També inicialitzem el timer *Clock1* per garantir que genera un esdeveniment cada dècima de segon.

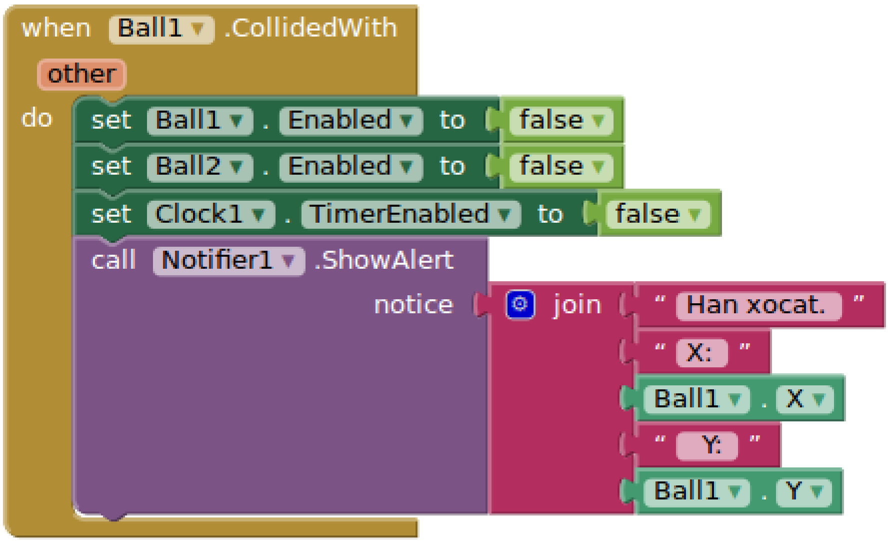
Cada esdeveniment que provocarà el *Clock1* serà detectat i en cada ocasió s'incrementarà en una dècima el valor de la variable global *cronom* i després s'assignarà el seu valor al *Label1* per mostrar el cronòmetre.



Cada cop que un sprite toqui a una de les quatre vores del *Canvas* on s'estan movent rebotaran en sentit contrari.




Quan hi hagi un xoc entre les boles, aquestes aturaran el seu moviment conjuntament amb el compteig del cronòmetre i es mostrarà una notificació a *Notifier1* indicat les coordenades X i Y del xoc.

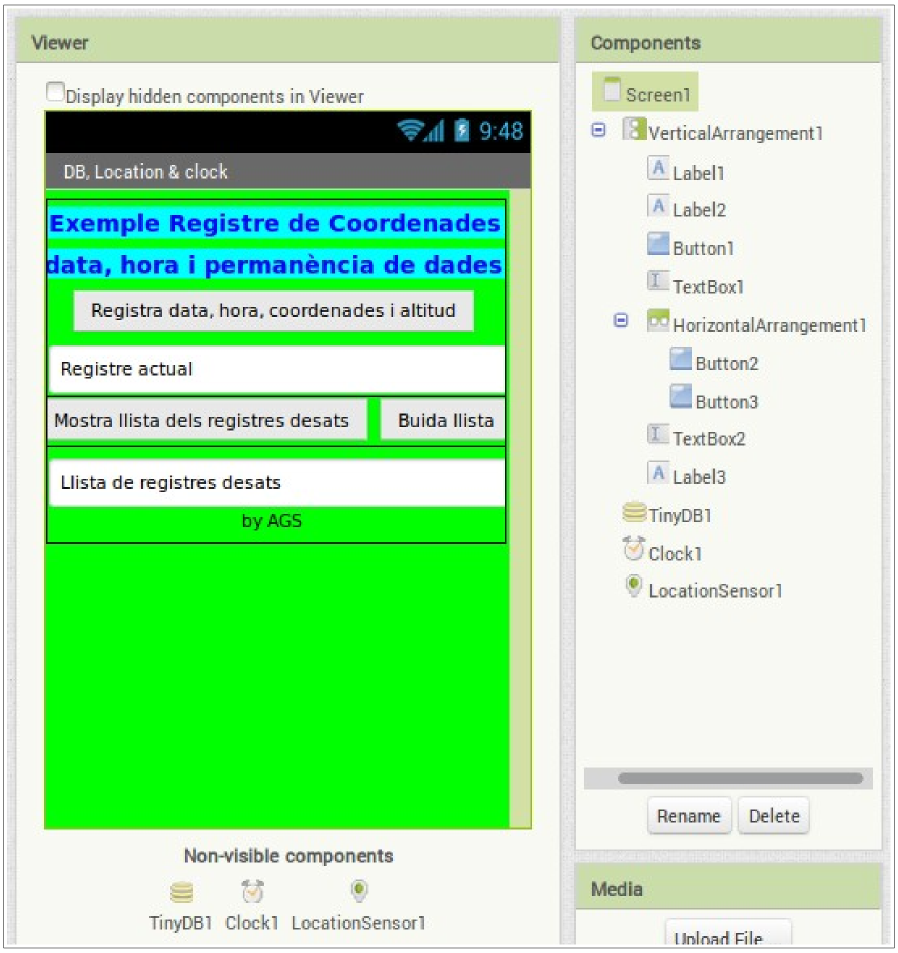


## Resultat:

5554:<build>
3:50 AM
Xocs cronometrats
Han xocat. X: 256.39001 Y: 174.04747
by AGS
Iniciar
2.5 segons
MENU

## Exercici 4. Solució:

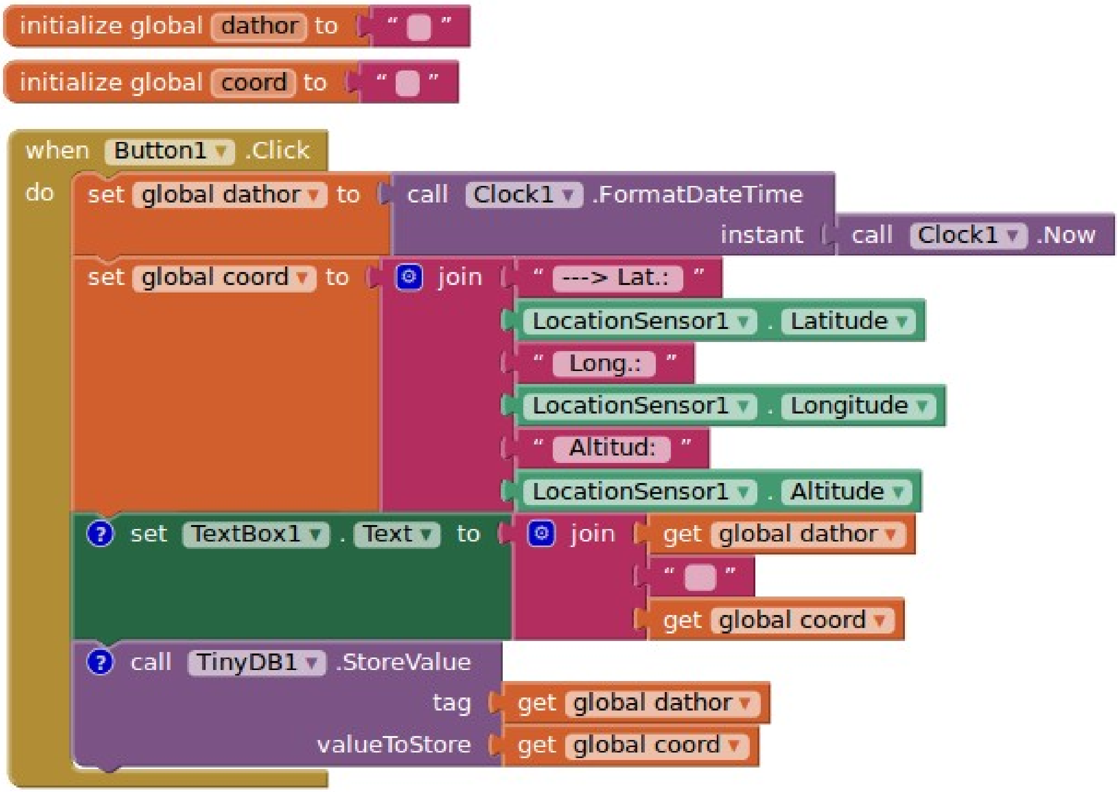
### Disseny:



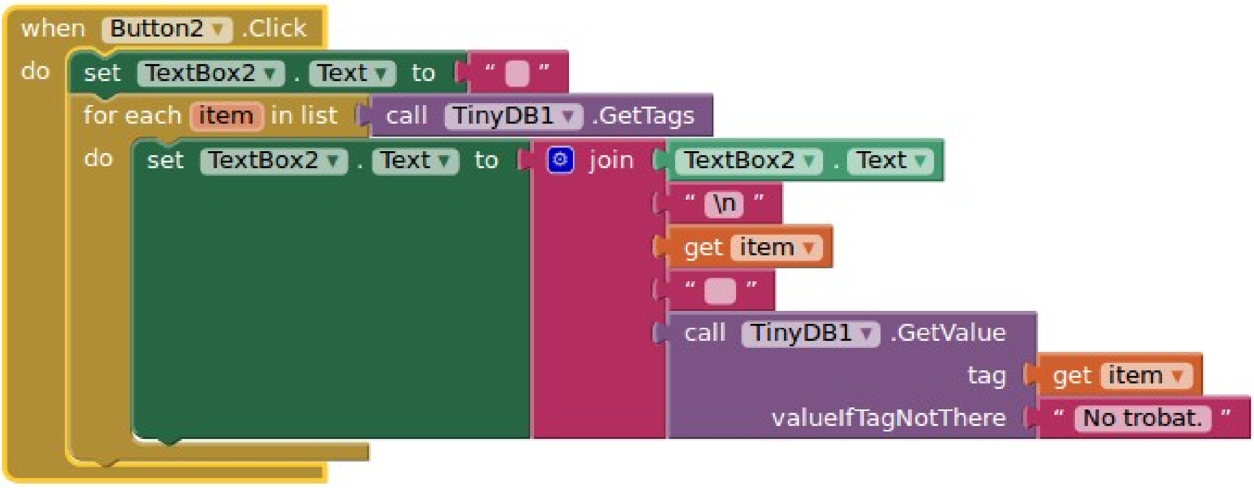

Utilitzem els components no visibles *Clock1* per poder disposar de la data i hora actuals, *LocationSensor1* per determinar les coordenades geogràfiques X, Y i la altitud i també utilitzem el component *TinyDb1* per implementar la persistència de dades i disposar-les encara que hàgim tancat el programa i el tornem a obrir. *TinyDb1* és una petita base de dades que permet desar-hi parelles tipus *clau=valor*, on, posteriorment podem recuperar els valors indicant el valor de la clau o recorrent totes les claus amb una estructura iteradora.

### Blocs:

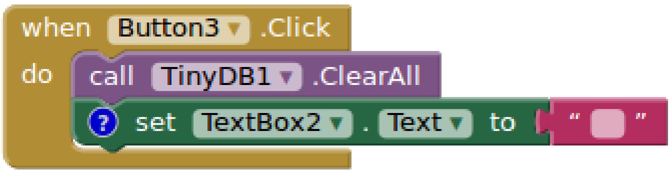
Inicialitzem les variables globals *dathor* per contenir la data i hora actual i *coord* per contenir les coordenades geogràfiques i l'altitud.
En clicar sobre el botó *Registra...* es fa la lectura de la data i hora actual des del mateix *Clock1* i de la posició geogràfica des del *LocationSensor1*.
Aquest valors són mostrats en el *TextBox1* i, a la vegada, són inserits a la base de dades *TinyDb1* on el contingut de *datahor* és la clau (tag) i *coord* és el valor a desar.
Les dades desades dins la base de dades no desapareixeran quan es tanqui el programa, es a dir, tindrem una persistència de dades, per tant, sempre estaran disponibles, a no ser que se'n faci un buidat exprés.



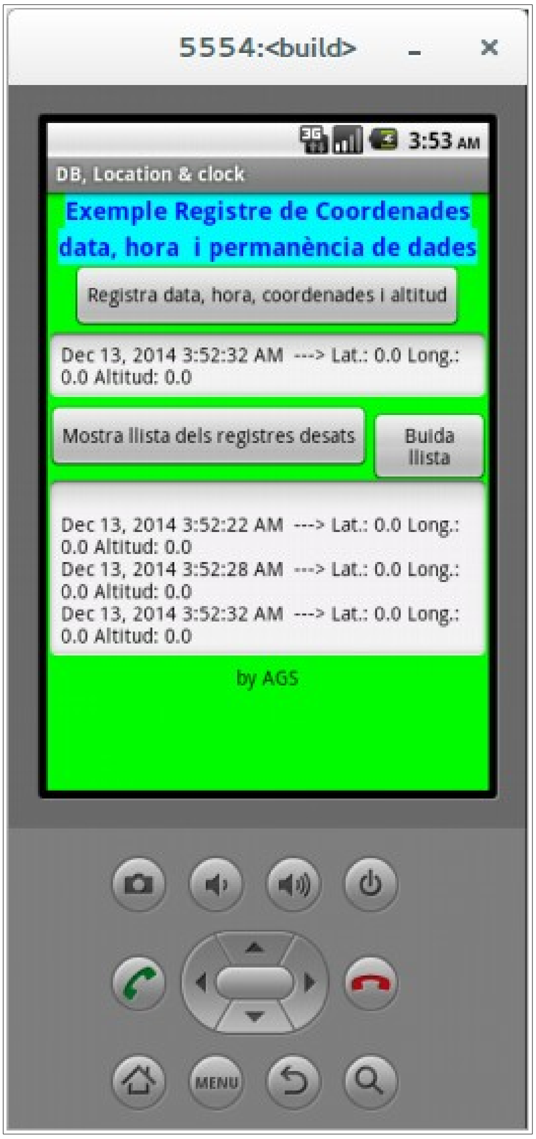

Per visualitzar el contingut de les dades desades a la base de dades, ho fem clicant sobre el botó *Mostra...*. Amb aquest procés es llegeixen totes les claus de la base de dades (GetTags) i després és fa una iteració amb aquestes claus extraient els valors (GetValue). Tota la informació es mostra en el *TextBox2*.



Finalment, si volem esborrar totes les dades desades a la base de dades, ho fem clicant a sobre del botó *Buida llista* que invoca el procés següent on es buida tota la base de dades i el contingut de *TextBox2*:




**Resultat:**



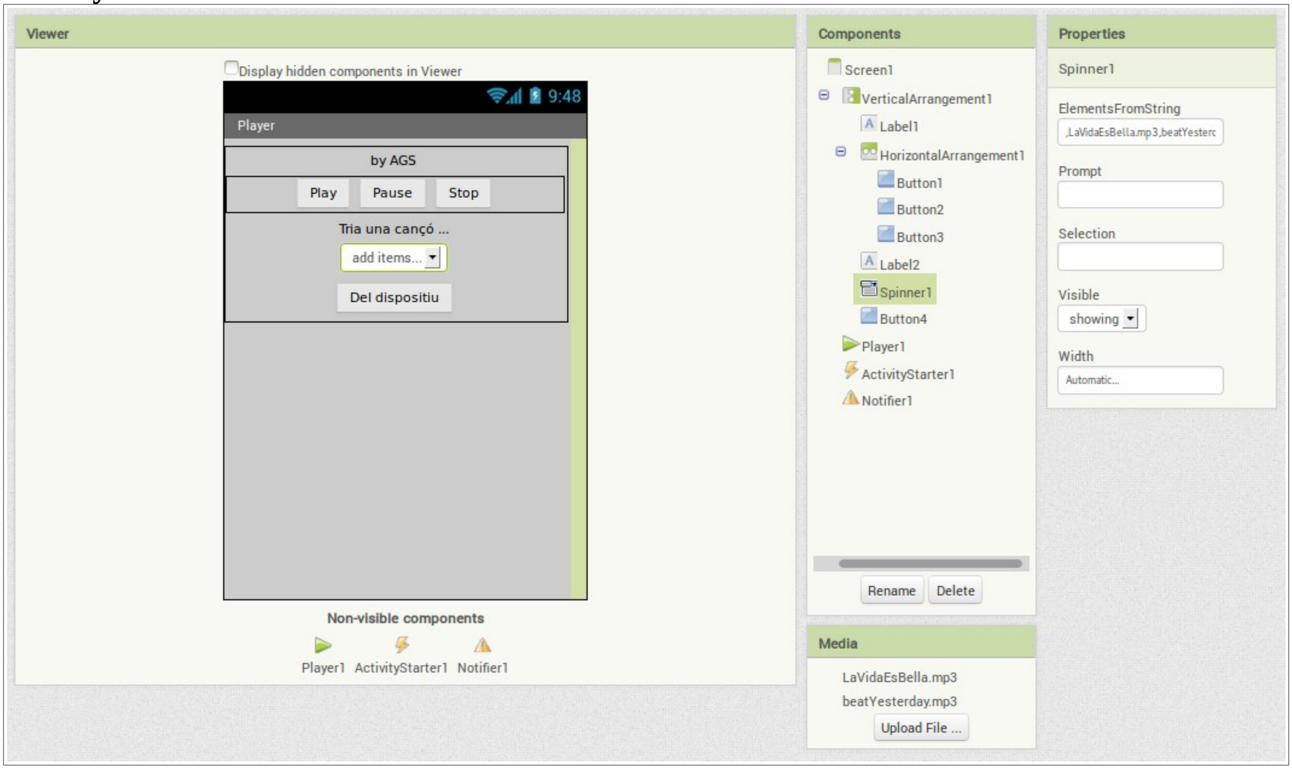
## Exercici 5. Solució:

La solució implementada permet que els fitxers de so a reproduir es puguin obtenir des de dos orígens possibles:

a) Una llista de fitxers de so carregats localment dins el mateix programa.

b) Seleccionar el fitxer de so dins el dispositiu smartphone on s'executa la app.

**Disseny:**

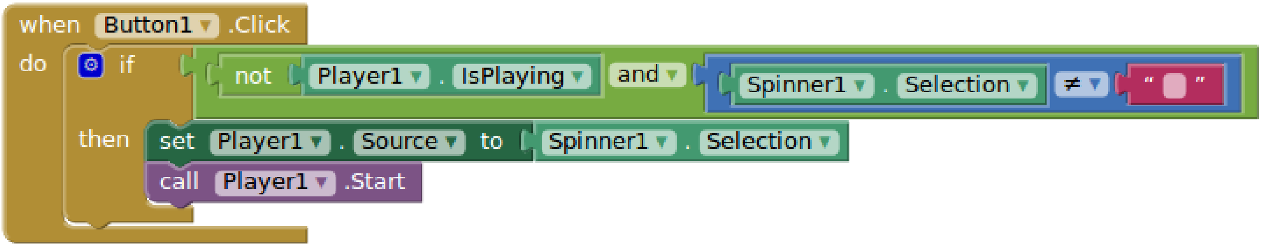



Utilitzem el component no visible *Player1* per reproduir la música.

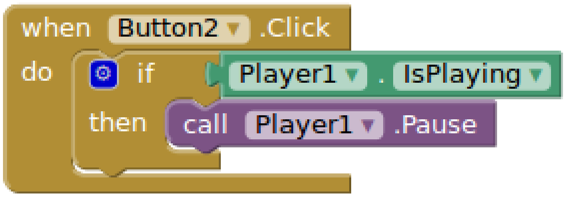
Per la opció a), la música s'haurà d'haver carregat al projecte en l'apartat Media. Els mateixos noms dels fitxers de so que hem carregat a Media els hem de posar, separats amb una coma, a la propietat: *ElementsFromString* del component *Spinner1*, en el nostre exemple, ha de ser el text: “LaVidaEsBella.mp3,beatYesterday.mp3”. Aquest component *Spinner1* contindrà la llista de cançons carregades en el mateix programa. Cal anar en compte amb aquesta opció, ja que pot fer que l'app tingui una mida excessiva si hi carreguem moltes cançons.

Per la opció b), elegirem la cançó a reproduir des d'una ubicació dins l'smartphone on s'executa la app. Per això usarem un component no visible *ActivityStarter1* que serveix per executar qualsevol programa extern a la nostra app. En el nostre cas, utilitzem l'*ActivityStarter1* per executar un programa extern que ens permeti triar una cançó dins el dispositiu smartphone. El component no visible *Notifier1* servirà per avisar en cas que no sigui possible trobar un programa instal·lat per triar la cançó.

**Blocs:**
Per al Play, per reproduir la cançó carregada dins la mateixa app, comprovem que hi hagi seleccionat un fitxer en el *Spinner1* i que el *Player1* no estigui reproduint cap cançó. Si es així, carregarà el fitxer indicat en el *Spinner1* i el reproduirà.



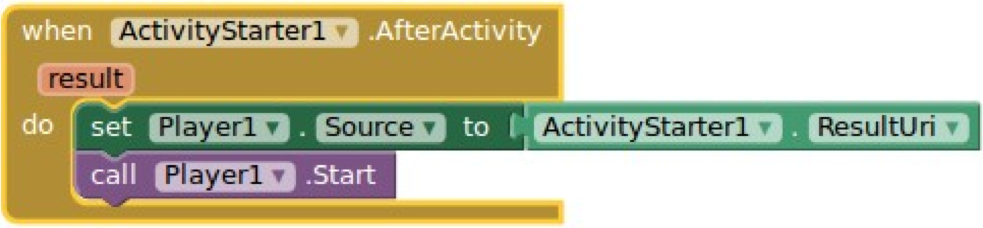

Per al Pause, comprovem que el *Player1* estigui reproduint un fitxer. Si es cert, podrem aplicar el Pause.




Si cliquem sobre Stop, s'atura el *Player1*.

Si desitgem seleccionar un fitxer de dins el nostre dispositiu smartphone, ho farem clicant a sobre del *Button4*: "Del dispositiu". En aquest cas, primer s'atura el *Player1* i s'esborra la selecció del *Spinner1*. Posteriorment, es prepara el component *ActivityStarter1* per fer la selecció de fitxers de música *mp3*, també es comprova que existeixi instal·lat un programa que permeti fer aquesta selecció (sinó, s'avisa amb un missatge al *Notifier1*) i si les condicions per resoldre l'activitat són correctes, s'executa l'activitat de selecció del fitxer de so invocant a *StartActivity*.

Quan ja ha estat feta la selecció del fitxer de so, s'activa l'esdeveniment *AfterActivity* i, en aquest, s'assigna la localització de la cançó al *Player1* i s'inicia la seva reproducció.



**Resultat:**

5554:<build>
9:52 AM
Player
by AGS
Play
Pause
Stop
Tria una cançó ...
beatYesterday.mp3
Del dispositiu
MENU

-------------------------------------------------------